Prof. Dr. Ingo Claßen
Studiengang Wirtschaftsinformtik
FB Informatik, Kommunikation und Wirtschaft

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Projektbericht

Titel	Evaluation von Pentaho Business Analytics für den Einsatz in der Lehre an der HTW Berlin
Autoren	Frederic Ruschke Tino Dietze Sebastian Rabus Cedric Xavier Tchoutouo Kougan

Inhaltsverzeichnis

Inhaltsverzeichnis II
Abbildungsverzeichnis IV
Tabellenverzeichnis VIII
Abkürzungsverzeichnis IX
1 Einleitung 1
1.1 Ziele und Umfang 1
1.2 Rahmenbedingungen und Vorgehen 1
1.3 Pentaho Business Analytics Community Edition 3
1.4 Evaluationsszenario 6
1.4.1 Business Domain und Grundlage der Quelldaten 6
1.4.2 Datenmodell und Erzeugung der Quelldaten 7
1.4.3 Star-Schema 14
2 BI-Server 19
2.1 Übersicht 19
2.2 Oberfläche und Bedienbarkeit 19
2.3 Funktionalität 21
2.4 Dokumentation 21
2.5 Umsetzung Szenario 21
3 ETL-Prozess 24
3.1 Übersicht 24
3.2 Oberfläche und Bedienbarkeit 24
3.3 Funktionalität 25
3.3.1 Input 26
3.3.2 Manipulation 30
3.3.3 Flusssteuerung 37
3.3.4 Datenprüfung und Fehlerbehandlung 39
3.3.5 Datenbanken 43
3.3.6 Job 48
3.4 Dokumentation 50
3.5 Umsetzung Szenario 50
3.5.1 Laden der Dimensionen 50
3.5.2 Laden der Faktentabelle 55
4 OLAP 58
4.1 4.1 Übersicht 58
4.2 Oberfläche und Bedienbarkeit 60
4.3 Funktionalität 61
4.3.1 Schema Editor 61
4.3.2 MDX-Queries 63
4.3.3 JDBC Explorer 64
4.4 Dokumentation 65
4.5 Umsetzung Szenario 66
4.5.1 Faktentabelle und Measures 66
4.5.2 Dimensionen und Hierarchien 67

4.5.3 Publishing auf Server ... 72
5 Reporting ... 73
5.1 Übersicht ... 73
5.2 Oberfläche und Bedienbarkeit ... 73
5.3 Funktionalität ... 76
5.4 Dokumentation ... 80
5.5 Umsetzung Szenario ... 80
5.5.1 Szenario 1 Erstellung einer Kundenliste ... 81
5.5.2 Szenario 2 Darstellungen mittels Diagramme ... 82
5.5.3 Szenario 3 Filtern mittels Parameter ... 84
6 Visualisierung und Dashboards ... 89
6.1 Übersicht ... 89
6.2 Oberfläche und Bedienbarkeit ... 90
6.3 Funktionalität ... 91
6.3.1 Datenauswahl ... 92
6.3.2 Modellierung ... 93
6.3.3 Ausgabe ... 95
6.4 Dokumentation ... 97
6.5 Umsetzung Szenario ... 97
6.5.1 Szenario 1 Filter ... 98
6.5.2 Szenario 2 Sort ... 101
6.5.3 Szenario 3 Limit ... 102
6.5.4 Szenario 4 PeriodsToDate ... 103
7 Data Mining ... 105
7.1 Übersicht ... 105
7.2 Funktionalität ... 106
7.3 Dokumentation ... 107
7.4 Umsetzung Szenario ... 108
8 Fazit ... 112
Anhang ... X
A0 Arbeitsumgebung ... X
A1 SQL Create Script ... XIII
A2 SQL Grundlage für Reports ... XIX
A3 XML-Struktur Mondrian Schema ... XXII
A4 Unterschiede zwischen der Enterprise und der Community Edition von Pentaho ... XXV
A5 Pentaho Data Integration Übersicht Modellierungsobjekte ... XXVI
A6 MDX-Queries ... XXVIII

Abbildungsverzeichnis

Abbildung 1 ER-Modell Webshop 7
Abbildung 2 Quelldatengenerierung Customer 8
Abbildung 3 Quelldatengenerierung Shipping company 9
Abbildung 4 Quelldatengenerierung Product 10
Abbildung 5 Quelldatengenerierung Delivery 11
Abbildung 6 Quelldatengenerierung Delivery_item 12
Abbildung 7 Quelldaten Geolocation 13
Abbildung 8 Physisches Datenmodell Star-Schema 14
Abbildung 9 Admin Console Oberfläche 19
Abbildung 10 User Console Oberfläche 20
Abbildung 11 PDI Oberfläche 25
Abbildung 12 PDI Text File Input Objekt Übersicht 27
Abbildung 13 PDI Text File Input Reiter Content 28
Abbildung 14 PDI Get System Data Objekt 29
Abbildung 15 PDI Auflistung Infotypes System Data Objekt 29
Abbildung 16 PDI Calculator Objekt 31
Abbildung 17 PDI Calculator Objekt Funktionen 32
Abbildung 18 PDI Formular Objekt Funktionen 33
Abbildung 19 PDI Group By Objekt 33
Abbildung 20 PDI Select Values Objekt 34
Abbildung 21 PDI Sort Rows Objekt 35
Abbildung 22 PDI Unique Rows Objekt 35
Abbildung 23 PDI Filter Rows Objekt 36
Abbildung 24 PDI Modellierungsobjekte zur Strukturänderung 36
Abbildung 25 PDI Data Movement Typen 38
Abbildung 26 PDI Data Validator Objekt 40
Abbildung 27 PDI Logging 41
Abbildung 28 PDI Step Metrics Logging 41
Abbildung 29 PDI Error Logging 42
Abbildung 30 PDI Performance Logging 42
Abbildung 31 PDI Data Explorer 43
Abbildung 32 PDI Table Input Objekt SQL Assistenz 44
Abbildung 33 PDI Table Output Objekt 45
Abbildung 34 PDI Dimension Lookup / Update Objekt 46
Abbildung 35 PDI Database Lookup Objekt 47

Abbildung 36 Transformation Dim_Customer 51
Abbildung 37 Transformation Dim_Geo 52
Abbildung 38 Transformation Dim_Mixed 52
Abbildung 39 Transformation Dim_Time 53
Abbildung 40 Transformation Dim_Date 53
Abbildung 41 Transformation Dim_Product 54
Abbildung 42 Transformation Fact_Delivery_item Extraktion 55
Abbildung 43 Transformation Fact_Delivery_item Lookups 56
Abbildung 44 Transformation Fact_Delivery_item Lookup Dim_Time 56
Abbildung 45 Transformation Fact_Delivery_item Kennzahlen und Speichern 57
Abbildung 46 Architektur Pentaho Mondrian 58
Abbildung 47 Übersicht Schema Workbench 60
Abbildung 48 PSW Übersicht Schema Editor 62
Abbildung 49 PSW Beispiel MDX-Query 63
Abbildung 50 PSW JDBC-Explorer 64
Abbildung 51 PSW Repository Login 72
Abbildung 52 PRD Startbildschirm 73
Abbildung 53 PRD Übersicht Arbeitsbereich 74
Abbildung 54 PRD Structure/Data Reiter 74
Abbildung 55 PRD Modellierungsbereich 75
Abbildung 56 PRD Connectionmanager 78
Abbildung 57 PRD Verbindungstypen 78
Abbildung 58 PRD Query Management 79
Abbildung 59 PRD Übersicht Edit Chart 79
Abbildung 60 PRD Szenario 1 Query 81
Abbildung 61 PRD Szenario 1 Reportentwurf 81
Abbildung 62 PRD Szenario 1 Ausgabe unformatiert 82
Abbildung 63 PRD Szenario 1 Ausgabe formatiert 82
Abbildung 64 PRD Ausgabe Szenario 2 Diagramme 83
Abbildung 65 PRD Query Szenario 2 83
Abbildung 66 PRD Query Szenario 3 Filter 84
Abbildung 67 PRD Reportentwurf Szenario 3 84
Abbildung 68 PRD Szenario 3 Verknüpfung des Import-Parameters 85
Abbildung 69 PRD Szenario 3 Parameterauswahl 85
Abbildung 70 PRD Szenario 3 Auswahlergebnis 85
Abbildung 71 PRD Szenario 3 MDX Query 86
Abbildung 72 PRD Szenario 3 MDX Ausgabe 87

Abbildung 73 PRD Szenario 3 SQL Ausgabe 88
Abbildung 74 Saiku Startbildschirm 90
Abbildung 75 Saiku Datenauswahl 92
Abbildung 76 Saiku Modellierungsbereich 93
Abbildung 77 Saiku MDX Editor 93
Abbildung 78 Saiku Hierarchieelement 93
Abbildung 79 Saiku Elementfilter 94
Abbildung 80 Saiku Filter/Limit/Sort Optionen 94
Abbildung 81 Saiku Custom Filter 95
Abbildung 82 Saiku Ausgabe 95
Abbildung 83 Saiku Diagrammausgabe 96
Abbildung 84 Saiku Szenario Ausgangsituation 97
Abbildung 85 Saiku Szenario 1 Elementfilter 98
Abbildung 86 Saiku Szenario 1 Ausgabe Elementfilter 98
Abbildung 87 Saiku Szenario 1 grafische Ausgabe Elementfilter 99
Abbildung 88 Saiku Szenario 1 Customfilter 99
Abbildung 89 Saiku Szenario 1 Ausgabe Customfilter 100
Abbildung 90 Saiku Szenario 1 Ausgabe Filter Kundentyp 100
Abbildung 91 Saiku Szenario 2 Sort 101
Abbildung 92 Saiku Szenario 2 Ausgabe 101
Abbildung 93 Saiku Szenario 3 Limit Ausgabe 102
Abbildung 94 Saiku Szenario 4 PeriodsToDate MDX 103
Abbildung 95 Saiku Szenario 4 Modeller 104
Abbildung 96 PDI Marketplace 106
Abbildung 97 PDI Weka Scoring Plugin Eigenschaften 106
Abbildung 98 PDI Weka Fehlermeldung 107
Abbildung 99 Pentaho Wiki / Weka Scoring Plugin 107
Abbildung 100 Weka Clustering Ausgabe 110
Abbildung 101 PDI Transformation Clustering 110
Abbildung 102 PDI Weka Scoring Objekt 111
Abbildung 103 PDI Weka Scoring Ausgabe 111
Abbildung 104 SQL Statement Berechnung Umsatz XIX
Abbildung 105 Ausgabe Berechnung Umsatz XIX
Abbildung 106 SQL Statement View Turnover_year XX
Abbildung 107 SQL-Statement Berechnung Gesamtumsatz je Jahr XX
Abbildung 108 Ergebnis Berechnung Gesamtumsatz je Jahr XX
Abbildung 109 SQL-Statement Pivot-Umwandlung XX

Abbildung 110 Ergebnis Pivot-Umwandlung XXI
Abbildung 111 SQL-Statement Delta-Berechnung und Prozentwerte XXI
Abbildung 112 Ergebnis Delta-Berechnung und Prozentwerte XXI
Abbildung 113 angelegte Views der Datenbank Webshop.....XXI

Tabellenverzeichnis

Tabelle 1 Funktionsunterschiede Community vs. Enterprise Edition 3
Tabelle 2 verwendete Komponenten 5
Tabelle 3 Quelldatenstruktur Customer 8
Tabelle 4 Quelldatenstruktur Shipping company 9
Tabelle 5 Quelldatenstruktur Product 10
Tabelle 6 Quelldatenstruktur Delivery 11
Tabelle 7 Quelldatenstruktur Delivery_item 12
Tabelle 8 Felder und Beschreibung fact_delivery_item 15
Tabelle 9 Felder und Beschreibung dim_product 15
Tabelle 10 Felder und Beschreibung dim_customer 16
Tabelle 11 Felder und Beschreibung dim_date 16
Tabelle 12 Felder und Beschreibung dim_time 17
Tabelle 13 Felder und Beschreibung dim_mixed 17
Tabelle 14 Felder und Beschreibung dim_geo 18
Tabelle 15 PDI Input-Modellierungsobjekte 26
Tabelle 16 PDI Manipulations-Modellierungsobjekte 30
Tabelle 17 PDI Filter/Sortierungs-Modellierungsobjekte 33
Tabelle 18 PDI Flusssteuerungs-Modellierungsobjekte 37
Tabelle 19 PDI Datenbank-Modellierungsobjekte 44
Tabelle 20 PDI Job-Modellierungsobjekte 48
Tabelle 21 Aggregatfunktionen der Measures 67
Tabelle 22 Attribute Produktdimension 67
Tabelle 23 Attribute Produkthierarchie 68
Tabelle 24 Attribute Produktlevel 68
Tabelle 25 Attribute Tabelle Produktdimension 69
Tabelle 26 Attribute der Lieferdimension 69
Tabelle 27 Attribute der default Hierarchie 70
Tabelle 28 Attribute des Lieferlevel 70
Tabelle 29 Attribute Lieferpositionlevel 71
Tabelle 30 PRD Übersicht Steuerelemente 76
Tabelle 31 Saiku Modellierungsobjekte 91

Abkürzungsverzeichnis

API	Application Programming Interface
CSV	Comma Separated Values
JAR	Java Archive
JDBC	Java Database Connectivity
MDX	Multidimensional Expressions
OLAP	Online Analytical Processing
PDI	Pentaho Data Integration
PGP	Pretty Good Privacy
PRD	Pentaho Report Designer
PSW	Pentaho Schema Workbench
RDBMS	Relational Database Management System
ROLAP	Relational Online Analytical Processing
SQL	Scripted Query Language
XML	Extensible Markup Language
XML/A	XML for Analysis

1 Einleitung

1.1 Ziele und Umfang

Im Masterstudiengang Wirtschaftsinformatik an der HTW Berlin werden zu Ausbildungszwecken verschiedene Datawarehouse und Business Intelligence Tools eingesetzt. Im Rahmen der Lehrveranstaltung Projekt Datawarehouse (P DWH) im SoSe 2013 bei Prof. Claßen soll die Open-Source Business Intelligence Suite von Pentaho auf ihre Eignung für den Lehreinsatz hin geprüft werden.

Dies umfasst insbesondere die Gebiete

- Extraktion, Transformation und Laden von Quelldaten (ETL),
- Multidimensionale Modellierung und Analyse (OLAP),
- Reporting und Dashboards,
- Data Mining

die in verschiedenen Lehrveranstaltungen behandelt werden.
Die bisher an der HTW Berlin in den o.g. Gebieten eingesetzten Programme sind meist kommerzielle Closed-source Anwendungen insbesondere von Microsoft und SAP. Das Projekt bzw. seine Ergebnisse sollen damit eine Entscheidungsgrundlage dafür schaffen, ob ein Open-Source Produkt in Teilbereichen zu Lehrzwecken genutzt werden könnte.

1.2 Rahmenbedingungen und Vorgehen

Das Projekt wird in weiten Teilen auf Infrastruktur der HTW durchgeführt, um eine leichte Adaption an die Lehrumgebung zu ermöglichen. Zu diesem Zweck wurde von Prof. Claßen eine virtuelle Maschine mit Windows 7 inkl. MS SQL Server 2008 bereitgestellt (Verbindungs- und Zugangsdaten siehe Anhang A0). Installation, Konfiguration und Betrieb von Pentaho werden auf dieser VM durchgeführt. Um eine parallele Bearbeitung zu ermöglichen, werden jedoch auch lokale Installationen genutzt. Als Referenzinstallation dient die bereitgestellte virtuelle Maschine, die nach Durchführung des Projekts den finalen Stand enthält. Um den Open Source Charakter des Projekts zu erhalten, wird als relationales Datenbankmanagementsystem MySQL in Version 5.6.11 eingesetzt.

Vor der Installation und Erprobung von Pentaho ist weiterhin das Aufstellen eines Evaluationsszenarios sowie die Generierung von passenden Testdaten notwendig. Auf dieser Grundlage können Kennzahlen definiert und ein Star-Schema modelliert werden.

Im weiteren Verlauf des Projekts werden die Extraktion, Transformation und das Laden der Quelldaten durchgeführt, um die Daten für das Star-Schema vorzubereiten und dieses zu befüllen. Dabei werden auch Besonderheiten wie Fehlerbehandlung und Slowly Changing Dimensions berücksichtigt.

Die physische Implementierung des Star-Schemas erfolgt in der MySQL-Datenbank. Darauf aufbauend wird ein OLAP-Schema bzw. Cube generiert, um mittels einer OLAP-Engine multidimensionale Analysen und visuelle Auswertungen durchführen zu können.
Abschließend wird die Funktionalität von Pentaho im Hinblick auf Reporting und Data Mining untersucht und ein Fazit bzgl. der Vor- und Nachteile sowie der Nutzbarkeit in der Lehre gezogen.

Um eine konsistente Dokumentation und bessere Lesbarkeit zu erreichen, wird ein standardisiertes Vorgehen bei der Evaluation von Pentaho genutzt. Zu Beginn werden die Plattform und die Komponenten vorgestellt. Im weiteren Verlauf werden die Client- und Server Tools anhand folgender Punkte auf verschiedene Eigenschaften untersucht:

- Übersicht (Plattform und Architektur, Deployment, Artefakte)
- Oberfläche und Bedienbarkeit (Screenshot “Startbildschirm”, Erläuterung der wichtigsten Bedienelemente, Konsistenz)
- Funktionalität (Umfang, Besonderheiten/Kernfunktionen)
- Dokumentation (“Hilfefunktion” im Tool, Verfügbarkeit und Vollständigkeit)
- Umsetzung des Szenarios

Unter dem Punkt “Umsetzung Szenario” wird die konkrete Nutzung des jeweiligen Tools im Hinblick auf das Evaluationsszenario beschrieben. Er ist daher immer der letzte Teil eines Kapitels.

Auf die Untersuchung von Performanceaspekten wird verzichtet, da diese unter den o.g. Rahmenbedingungen (Nutzung einer limitierten virtuellen Maschine) nicht objektiv bewertet werden können.

1.3 Pentaho Business Analytics Community Edition

Bei der Pentaho Business Analytics Suite handelt es sich um eine Business-Intelligence Software welche die Bereiche ETL, Reporting, OLAP und Data-Mining abdeckt. Sie wird seit 2004 von der Pentaho Corporation in Orlando, Florida, USA entwickelt. Die gesamte Software ist in Java geschrieben und bietet eine Webanwendung als Frontend für den Endanwender, wodurch eine Plattformunabhängigkeit gegeben ist. Die einzelnen Komponenten der Anwendung sind aus verschiedenen Open-Source-Projekten hervorgegangen. Pentaho ist daher sowohl kostenlos und quelloffen unter der Apache-Lizenz (Version 2.0), als auch als kommerzielle Version mit Support und größerem Funktionsumfang verfügbar (vgl. http://de.wikipedia.org/wiki/Pentaho und http://en.wikipedia.org/wiki/Pentaho).

Die folgende Tabelle listet einige der Funktionsunterschiede zwischen den Versionen in Englisch auf:

Tabelle 1 Funktionsunterschiede Community vs. Enterprise Edition

Feature	**Community Edition**	**Enterprise Edition**
Automated Installers	No	Yes
Sophisticated & Interactive Data Analysis, Exploration & Visualization	“Basic analysis”	Yes
Self-Service Dashboard Designer	No	Yes
Support for Mobile BI (e.g. iPad)	No	Yes
Shared Repository and Team Sharing	No	Yes
User and Role-based Permissions	No	Yes
Security Configuration with LDAP & Active Directory	No	Yes
Single Sign-On	No	Yes

Automated ETL Job Scheduling	No	Yes
Centralized Administration, Diagnostics, and Performance Monitoring	No	Yes

Für eine vollständige Auflistung der Unterschiede zwischen den Versionen siehe Anhang A4.

Neben dem größeren Funktionsumfang besitzt die Enterprise Edition eine Installationsroutine welche die initiale Konfiguration und Einrichtung sämtlicher Komponenten automatisiert durchführt. Weiterhin enthält die Enterprise Edition sämtliche Komponenten in einer einzigen Installationsdatei. Insbesondere Werkzeuge zur grafischen Visualisierung und Auswertung (z. B. Dashboard) sind in der Community Edition nicht verfügbar. Gleiches gilt für Funktionen zur verteilten Zusammenarbeit (Repository), Zugriffskontrollen durch Rechte und Rollen sowie Administration und Monitoring. Alle diese Features sind lediglich in der Enterprise Edition vollständig implementiert bzw. nutzbar.
Da für die Durchführung des Projekts ausschließlich die Community Edition von Pentaho genutzt werden soll, können die o. g. Funktionen nicht oder nur in Ansätzen evaluiert werden. Auf die daraus resultierenden Einschränkungen wird bei Bedarf in den jeweiligen Kapiteln näher eingegangen.

Im Gegensatz zur Enterprise Edition besteht die Community Edition aus einer losen Ansammlung von Server- und Clienttools, die einzeln von der Projekthomepage heruntergeladen, ggf. installiert, konfiguriert und miteinander integriert werden müssen.
Im Rahmen des Projekts werden folgende Komponenten der Community Edition verwendet:

Tabelle 2 verwendete Komponenten

Komponente	**Dateiname inkl. Version**	**Funktionalität**
Business Intelligence Server (inkl. Mondrian)	biserver-ce-4.8.0.zip	Frontend, Administration, Integration, OLAP-Engine
Data Integration	pdi-ce-4.4.0.zip	ETL
Schema Workbench	psw-ce-3.5.0.zip	OLAP Modelling
Saiku Analytics	Version 2.5, aus Pentaho Marketplace	MDX, Analyse
Report Designer	prd-ce-3.9.0.zip	Reporting
Weka	weka-3-6-10.zip	Data Mining

Der Business Intelligence Server inkl. der OLAP-Engine Mondrian stellt laut Pentaho eine BI Plattform dar. Data Integration, auch bekannt als “Kettle”, deckt ETL ab. Die Schema Workbench und das Saiku Analytics Plugin sind der multidimensionalen Modellierung und Auswertung zuzuordnen. Der Report Designer und Weka dem Reporting bzw. Data Mining. Eine weitergehende Beschreibung der Tools findet sich im jeweiligen Kapitel in diesem Dokument.

Die genannten, sowie weitere Komponenten der Community Edition können unter folgenden Links heruntergeladen werden:
http://wiki.pentaho.com/display/COM/Getting+Started+Downloads
http://wiki.pentaho.com/display/COM/Latest+Stable+Builds

Die Verzeichnisstruktur der Komponenten auf der virtuellen Maschine ist in Anhang A0 dargestellt.

1.4 Evaluationsszenario

1.4.1 Business Domain und Grundlage der Quelldaten

Als Business Domain wird ein Handelsunternehmen zugrunde gelegt, welches Lebensmittel über das Internet verkauft. Es handelt sich also um ein “Webshop”-Szenario: Das Handelsunternehmen kauft Artikel bei seinen Lieferanten und verkauft diese mit Gewinn an Kunden weiter. Es wird lediglich der Verkauf von Waren betrachtet. Der Einkauf wird aus Gründen der Vereinfachung nicht berücksichtigt.

Die Kerntransaktion im Szenario ist dabei die Lieferung von bestellten Artikeln. Diese Artikel werden durch ein Transportunternehmen gegen eine Gebühr an die jeweilige Kundenadresse ausgeliefert. Bestelldatum und Lieferdatum werden getrennt gespeichert.

Das zugrunde liegende Szenario wurde bewusst “einfach” gewählt, da der Fokus des Projekts auf der Erprobung der grundlegenden Funktionalität der Pentaho BI-Suite liegt. Sämtliche Bezeichner der Quelldaten sind auf Englisch gewählt.

Die Quelldaten sollen mittels frei verfügbaren online Generatoren und unter Hilfe von Excel Funktionen erstellt werden. Dabei wurden folgende drei Generatoren genutzt:

- http://tedagen.com/
- http://www.mobilefish.com [1]
- http://migano.de

tedagen.com - Test Data Generator
tedagen.com generiert zum einen spezifische Daten aus den Themengebieten “IT & Telecom”, “Business & Finance” und “People & Localisation”, zum anderen Daten in diversen String und Number Formaten. Zusätzlich bietet tedagen noch weitere Operatoren zur Manipulation der Daten an. Eine Limitierung der generierten Zeilen ist nicht vorhanden.
Der Output kann u. a. in XML und SQL Format stattfinden, aber auch in den Sprachen Javascript, Perl, Ruby, JSON, PHP. Die Zielgruppe dieses Generators ist somit eher im Programmierungsumfeld.

[1] Siehe: http://www.mobilefish.com/services/random_test_data_generator/random_test_data_generator.php

Mobilefish.com

Mobilefish.com generiert Kunden-, Orts-, Finanz-, Firmen- und Produktdaten auf Basis bestehender Listen. Die Orts-, Finanz- und Kundendaten beziehen sich auf holländische und amerikanische Daten. Zusätzlich ist es möglich eigene Format Strings zu definieren. Die Ausgabe ist auf 5000 Zeilen limitiert und erfolgt im .csv oder Excel Format. Dieser Generator ist explizit dafür ausgelegt Testdaten für Datenbanken zu erstellen.

migano.com

migano.com generiert Personen- und Ortsdaten aus dem deutschen Sprachraum. Im Gegensatz zu den anderen Generatoren lässt sich auch der Füllgrad der Spalten definieren. Die Ausgabe wird auf 1000 Zeilen limitiert und erfolgt im .csv Format.

1.4.2 Datenmodell und Erzeugung der Quelldaten

Aus dem oben genannten Konzept resultiert ein einfaches Datenmodell mit 5 Tabellen. Diese Daten sollen am Ende der Generierung als .csv Dateien vorliegen. Eine eigene Speicherung in einer Datenbank erfolgt somit nicht.

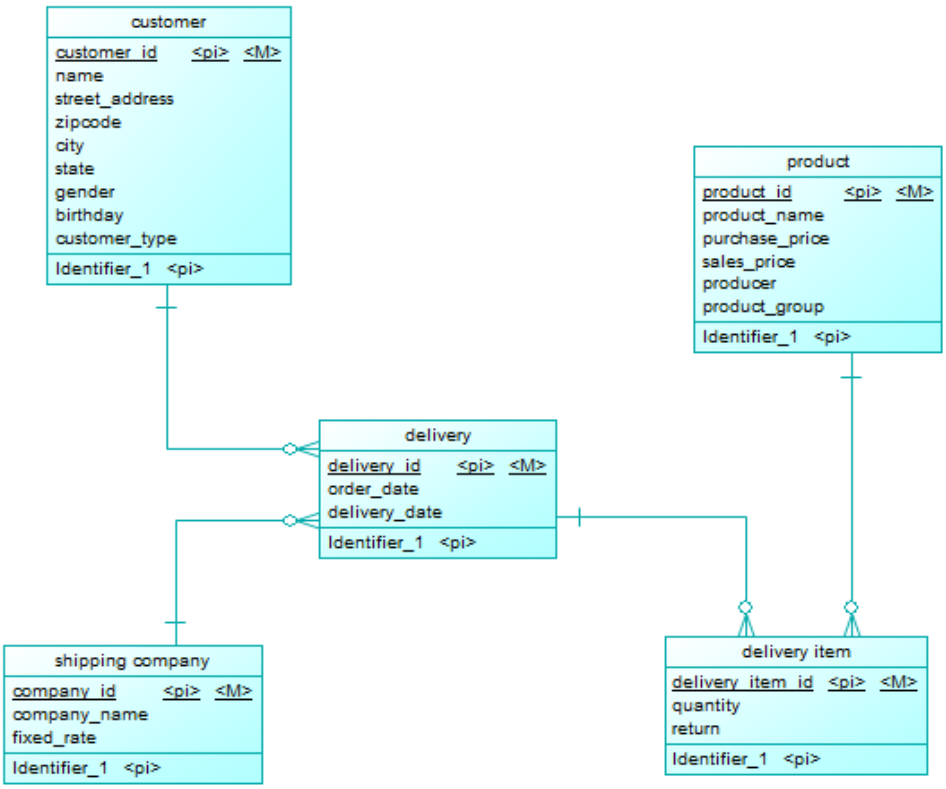


Abbildung 1 ER-Modell Webshop

Customer

Die Generierung der Ortsdaten erfolgte unabhängig von den Personendaten, um die Verteilung der Standorte selbst steuern zu können. So sind in eventuellen späteren Analysen Ergebnisse besser interpretierbar. Es wurden 2000 Kundendaten, mit 27 zugehörigen Bundesstaaten generiert. Circa 30% der Kunden haben den Kundentyp “1”. Das Geschlecht ist gleichverteilt. Die Kombination zipcode, city und state ist real existierend. Alle sonstigen Daten sind fiktiv oder stehen in keinen Zusammenhang.

Tabelle 3 Quelldatenstruktur Customer

Spalte	**Beschreibung**	**Datentyp**	**Wertebereich/Format**
Customer_ID	Kundennr	Integer	>100000000
Name	Kundennachname	String	255 Zeichen
Street_Address	Adresse	String	255 Zeichen
zipcode	Postleitzahl	String	5 Zeichen
city	Stadt	String	255 Zeichen
state	Bundesstaat	String	255 Zeichen
gender	Geschlecht	String	255 Zeichen
birthday	Geburtstag	date	YYYY-MM-DD
customertyp	Kundentyp	String	0;1

	A	B	C	D	E	F	G	H	I
1	Customer_ID	Name	Street_Addr	zipcode	city	state	Gender	birthday	customert
2	1000000001	Galvan	P.O. Box 155	94029	Menlo Park	California	F	1956-12-29	0
3	1000000002	Frey	P.O. Box 491	94030	Millbrae	California	M	1953-12-24	1
4	1000000003	Estes	25 E Narrow	94031	Millbrae	California	M	1953-04-05	0
5	1000000004	Snijders	441 Cotton S	94037	Montara	California	F	1972-07-01	1
6	1000000005	Mcdowell	P.O. Box 782	94038	Moss Beach	California	M	1941-05-12	1

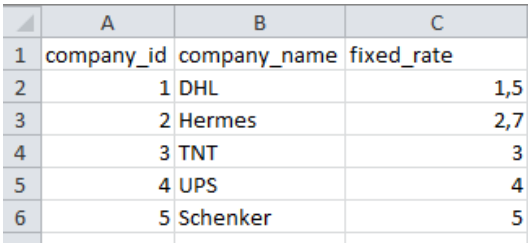
Abbildung 2 Quelldatengenerierung Customer

Shipping company

Es wurden 5 Versandunternehmen in Excel erstellt. Die Versandkostenpauschale wurde zur einfacheren Nutzung als Prozentwert hinterlegt. Zukünftige Berechnungen der Versandkosten erfolgen somit als prozentualer Anteil vom Umsatz.

Tabelle 4 Quelldatenstruktur Shipping company

Spalte	Beschreibung	Datentyp	Wertebereich/Format
company_id	Nummer	Integer	###
company_name	Name	String	255 Zeichen
fixed_rate	Versandkostenpauschale in %	Numeric	##,##



	A	B	C
1	company_id	company_name	fixed_rate
2	1	DHL	1,5
3	2	Hermes	2,7
4	3	TNT	3
5	4	UPS	4
6	5	Schenker	5

Abbildung 3 Quelldatengenerierung Shipping company

Product

In den Produktdaten sind neben dem Namen und der ID auch der Ein- und Verkaufspreis, sowie Produzent und Produktgruppe hinterlegt. Automatisch generiert wurden der Produktname und der Einkaufspreis. Die Produkt_id wurde als laufende Nummer im Excel erzeugt. Der Verkaufspreis wurde in Excel als das 1,05 bis 1,5-fache des Einkaufspreises berechnet. Ein negativer Profit ist somit nicht möglich. Bei der Generierung der Product_name Spalte wurde bereits eine Produktgruppe generiert, zu erkennen am "-" im Namen. Da nicht alle Namen diese enthalten wurde die Spalte product_group erstellt und mit 5 Produktgruppen gefüllt. Die Hersteller- und Produktgruppenwerte wurden zunächst in Excel erfasst und dann auf alle Produkte gleichverteilt. Insgesamt wurden 450 Produkte erzeugt, mit 10 Herstellern und 5 Produktgruppen.

Tabelle 5 Quelldatenstruktur Product

Spalte	Beschreibung	Datentyp	Wertebereich/Format
product_id	Produktnummer	Integer	>1000
product_name	Produktname	String	255 Zeichen
purchase_price	Einkaufspreis	Numeric	xxxxxxxx,xx
sales_price	Verkaufspreis	Numeric	xxxxxxxx,xx
producer	Hersteller	String	255 Zeichen
product_group	Produktgruppe	String	255 Zeichen

	A	B	C	D	E	F
1	product_id	product_name	purchase_pric	sales_price	producer	product_group
2	1000	7up Diet, 355 Ml	10,27	14,79	Nestle	Group_four
3	1001	Alize Red Passion	223,85	235,04	Anheuser-Bu	Group_five
4	1002	Anisette - Mcguiness	13,32	18,25	Dean Foods	Group_three
5	1003	Appetizer - Chicken Satay	133,33	149,33	Smithfield Fo	Group_three
6	1004	Appetizer - Crab And Brie	37,14	48,65	Smithfield Fo	Group_two
7	1005	Appetizer - Mango Chevre	33,90	45,43	Dean Foods	Group_five

Abbildung 4 Quelldatengenerierung Product

Delivery

Die Lieferdaten wurden mittels Excel Funktionen generiert. Alle ID's wurden als laufende Zahlen erstellt. Es gibt ca. 10.000 Lieferungen für das Jahr 2011 und ca. 5.000 für 2012. Die Uhrzeiten wurden mittels der Excel Verteilungsfunktion zufällig verteilt. Das Lieferungsdatum erfolgt immer zwischen ein und sechs Tage nach dem Bestelldatum, zu einer zufälligen Uhrzeit.

Tabelle 6 Quelldatenstruktur Delivery

Spalte	Beschreibung	Datentyp	Wertebereich/Format
delivery_id	Lieferungsnummer	Integer	>3000
customer_id	Kundennummer	Integer	>1000000000
shipping_company_id	Versandunternehmennummer	Integer	###
order_date	Bestelldatum	datetime	dd.mm.yyyy. hh:mm:ss
delivery_date	Lieferdatum	datetime	dd.mm.yyyy. hh:mm:ss

	A	B	C	D	E
1	delivery_id	customer_id	shipping_company_id	order_date	delivery_date
2	3000	1000001419	1	01.01.2010 04:20:58	04.01.2010 18:34:54
3	3001	1000001324	4	01.01.2010 16:20:16	05.01.2010 01:25:54
4	3002	1000000222	1	01.01.2010 07:12:14	03.01.2010 18:28:11
5	3003	1000000252	2	01.01.2010 16:49:07	05.01.2010 05:34:58
6	3004	1000001144	3	01.01.2010 12:51:21	02.01.2010 00:12:40
7	3005	1000001884	4	01.01.2010 14:48:23	04.01.2010 12:56:58

Abbildung 5 Quelldatengenerierung Delivery

Delivery_item

Die Generierung der Lieferpositionen erfolgte mittels Excel Funktionen. Es wurden ca. 100.000 Lieferpositionen angelegt. Eine Lieferung enthält zwischen einer und 19 Positionen und Pro Position zwischen ein und 10 Artikeleinheiten. Circa 10% aller Positionen sind Retourpositionen.

Tabelle 7 Quelldatenstruktur Delivery_item

Spalte	Beschreibung	Datentyp	Wertebereich/Format
delivery_id	Liefernummer	Integer	>3000
delivery_item_id	Positionsnummer	Integer	1-19
product_id	Produktnummer	Integer	>1000
quantity	Stückzahl	Integer	1-10
retour	Retour	Integer	0;1

	A	B	C	D	E
1	delivery_id	delivery_item_id	product_id	quantity	retour
2	3000	9	1325	4	0
3	3000	8	1114	1	0
4	3000	7	1343	1	0
5	3000	6	1226	8	0
6	3000	5	1448	8	0
7	3000	4	1158	2	0
8	3000	3	1341	3	0
9	3000	2	1263	2	1
10	3000	1	1326	6	0
11	3001	8	1367	9	0
12	3001	7	1132	4	0
13	3001	6	1169	4	0
14	3001	5	1047	10	0
15	3001	4	1046	6	0
16	3001	3	1431	2	0
17	3001	2	1066	1	1
18	3001	1	1103	2	0
19	3002	9	1400	6	0

Abbildung 6 Quelldatengenerierung Delivery_item

Geolocation

Zur Zuweisung von Postleitzahlen zu Latitude/Longitude wurde von der Website www.geonames.org eine Textdatei mit allen Postleitzahlen und zugehörigen Ortsdaten der USA bezogen.

```
1  US  34050   FPO      AA  Erie     029          41.0375 -111.6789
2  US  34034   APO      AA  Dillon   033          33.0364 -82.2493
3  US  99553   Akutan   Alaska  AK  Aleutians East  013          54.143  -165.7854
4  US  99571   Cold Bay     Alaska  AK  Aleutians East  013          55.3976 -162.4206
5  US  99583   False Pass   Alaska  AK  Aleutians East  013          54.841  -163.4368
6  US  99612   King Cove    Alaska  AK  Aleutians East  013          55.0628 -162.3056
7  US  99661   Sand Point   Alaska  AK  Aleutians East  013          55.3192 -160.4914
```

Abbildung 7 Quelldaten Geolocation

1.4.3 Star-Schema

Um verschiedene analytische Fragestellungen im Kontext des Evaluationszenarios beantworten zu können, wurde ein Star-Schema mit mehreren Dimensionen und Kennzahlen modelliert, dessen physisches Datenmodell im Folgenden dargestellt näher erläutert wird. Primär- und Fremdschlüsselfelder werden nicht näher erläutert, sondern können der Abbildung des physischen Datenmodells entnommen werden.

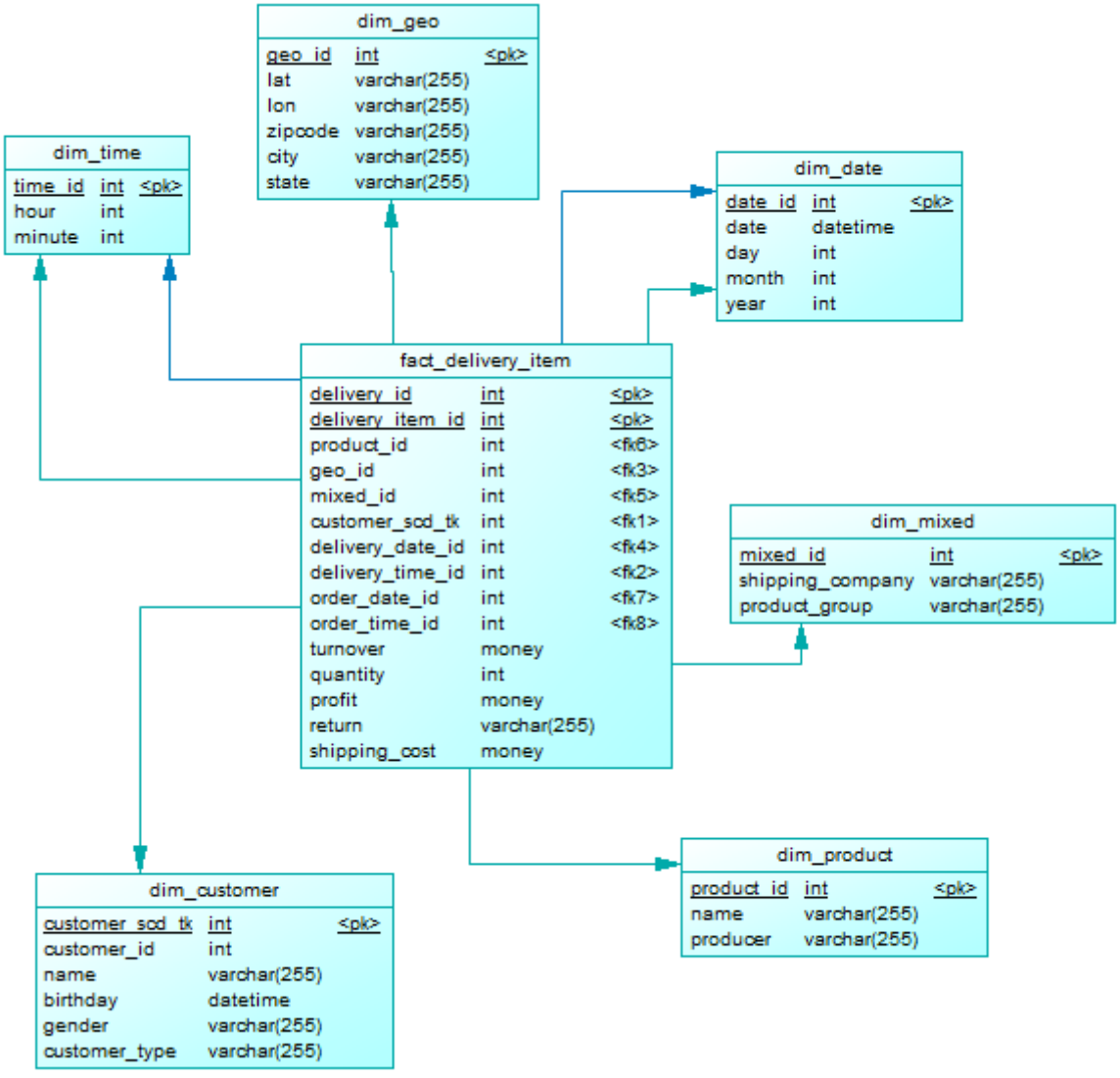


Abbildung 8 Physisches Datenmodell Star-Schema

Faktentabelle (fact_delivery_item)

Ein Eintrag in der Faktentabelle repräsentiert eine Lieferposition. Es wurde somit die feinstmögliche Granularität gewählt. Dies ermöglicht eine große Flexibilität, da einerseits die Bestandteile einer Lieferung auf Positionsebene analysiert werden können, andererseits mittels Aggregation auf eine Lieferung verdichtet werden kann.

Tabelle 8 Felder und Beschreibung fact_delivery_item

Feldname	Beschreibung
turnover	Der Umsatz der Lieferung: Umsatz = Menge * Verkaufspreis
quantity	Die Liefermenge des Produkts
profit	Gewinn aus dem Verkaufsvorgang, der sich wie folgt berechnet: ((Verkaufspreis - Einkaufpreis - Transportkosten) *Bestellmenge). Annahme: Der Versand an den Kunden ist immer kostenlos und schmälert daher den Gewinn.
return	Gibt an, ob die Lieferposition zurückgesendet wurde: true = ja, false = nein.
shipping_cost	Die anteiligen Transportkosten je Lieferposition.

Dimensionstabelle Produkt (dim_product)

Die Produktdimension referenziert das Produkt der jeweiligen Lieferposition.

Tabelle 9 Felder und Beschreibung dim_product

Feldname	Beschreibung
name	Produktname
producer	Hersteller des Produkts

Dimensionstabelle Kunde (dim_customer)

Die Kundendimension referenziert den Kunden der jeweiligen Lieferposition. Aufgrund der gewählten Granularität ist der Kunde sämtlicher Lieferpositionen einer Lieferung identisch.

Tabelle 10 Felder und Beschreibung dim_customer

Feldname	Beschreibung
name	Nachname des Kunden
birthday	Geburtstag des Kunden
gender	Geschlecht (male, female)
customer_type	Kundentyp (private, business)

Dimensionstabelle Datum (dim_date)

Erster Teil der Zeitbetrachtung auf Datumsebene. Wird sowohl für das Bestell- als auch das Lieferdatum in der Faktentabelle referenziert.

Tabelle 11 Felder und Beschreibung dim_date

Feldname	Beschreibung
date	Das zusammengesetzte Datum im Format YYYY-MM-DD HH:MM:SS
day	Nr. des Tages
month	Nr. des Monats
year	Jahr

Dimensionstabelle Zeit (dim_time)

Zweiter Teil der Zeitbetrachtung auf Ebene der Uhrzeit. Wird sowohl für den Zeitpunkt der Bestellung als auch für den Zeitpunkt der Lieferung in der Faktentabelle referenziert.

Tabelle 12 Felder und Beschreibung dim_time

Feldname	Beschreibung
hour	Stunde (HH)
day	Minute (MM)

Dimensionstabelle Mixed (dim_mixed)

Sammeldimension, die sowohl das Versandunternehmen als auch die Artikelgruppe referenziert.

Tabelle 13 Felder und Beschreibung dim_mixed

Feldname	Beschreibung
shipping_company	Name des Transportunternehmens
product_group	Name der Produktgruppe

Dimensionstabelle Geodaten (dim_geo)

Dient der Speicherung von Geoinformationen in Bezug auf den Wohnort des Kunden, d. h. des Lieferortes.

Tabelle 14 Felder und Beschreibung dim_geo

Feldname	**Beschreibung**
lat	Breitengrad
lon	Längengrad
zipcode	Postleitzahl
city	Stadt
state	Bundesland

Das Starschema wird in der MySQL-Datenbank deployed, um als Zielstruktur für die Daten aus dem ETL-Prozess zu dienen. Weiterhin stellt es die Grundlage für die multidimensionalen Auswertungen mittels der OLAP-Engine dar. (SQL Script siehe Anhang A1)

2 BI-Server

2.1 Übersicht

Der Bi-Server stellt die serverseitigen Tools zur Präsentation und Administration dar. Dies geschieht über die Portale User Console (Präsentation) und Admin Console (Administration). Die User Console stellt die Zugriffebene für den Endanwender dar. Dort hat er Zugriff zu Reporting- und Dashboardfunktionen. Die Admin Console bietet eine Rechteverwaltung und Monitoring Funktionen. Als Infrastruktur wird ein Apache Tomcat Server verwendet.

2.2 Oberfläche und Bedienbarkeit

Admin Console

Die Oberfläche der Admin Console ist schlicht in Menüband und Verwaltung getrennt. Über der Verwaltung wird grafisch der Serverstatus angezeigt. Mittels den Buttons Rechts oben lassen sich die Anzeigen Aktuallisieren und die Hilfe aufrufen.

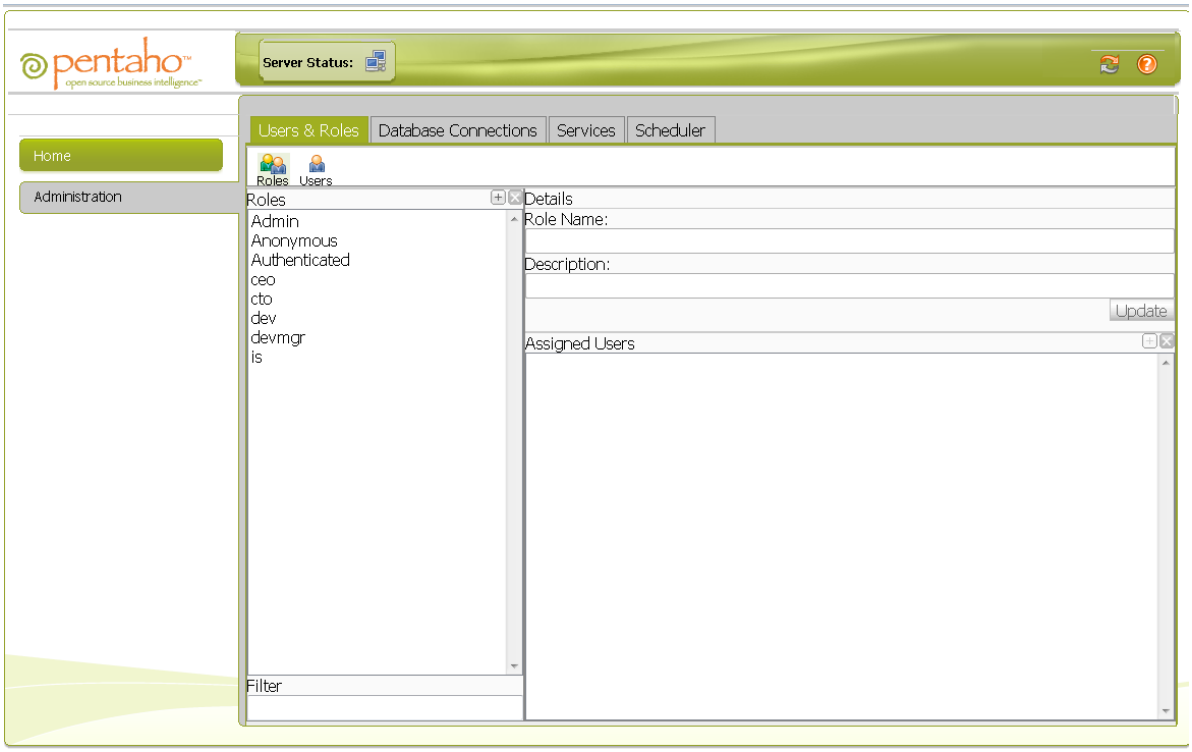


Abbildung 9 Admin Console Oberfläche

User Console

Nach Log-in erwartet den User ein Welcome Screen, der direkt eine Erstellung von neuen Reports oder Analysen bietet. Auf der Linken Seite werden die dem User freigegebene Ordnerstrukturen angezeigt. In diesen liegen die Reports/Dashboards. Die Menüleiste ermöglicht den schnellen Zugriff auf die wichtigsten Funktionen, wie Durchsuchen der Ordner, neue Dateien anlegen, Verwaltung des eigenen Arbeitsbereichs und Zugriff auf den Marketplace (mit Administratorrechten).

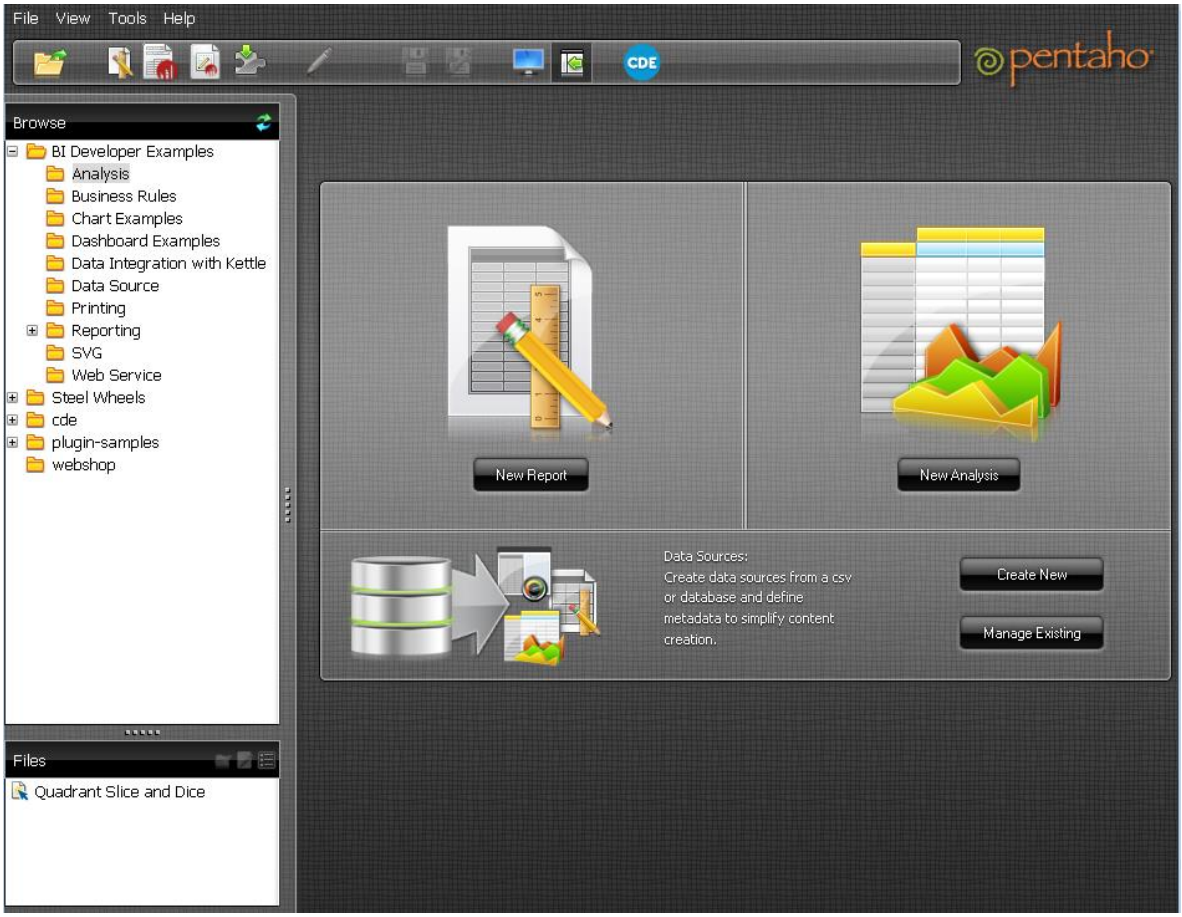


Abbildung 10 User Console Oberfläche

Im Marketplace lassen sich von der Community oder externen Anbietern Plugins einspielen. Derzeit sind nur wenige Plugins der Anbieter “WebDetails”, “Analytical Labs” und “Openl” vorhanden. Der größte Anbieter “WebDetails” wurde im April 2013 von Pentaho übernommen und die Tools wurden in die Enterprise Edition 5.0 integriert.

2.3 Funktionalität

Mittels der Admin Console können verwendete Server überwacht und Verwaltungsdienste gesteuert werden. In der Community Edition sind die Möglichkeiten sehr stark eingeschränkt. Es ist lediglich eine Userverwaltung und ein Serverstatus vorhanden.

Die User Console bietet in der Community Edition keine Tools zur Erstellung von Dashboards an. Zur Erstellung von Reports und Analysen werden jeweils Tools mitgeliefert, die bereits nicht mehr offiziell unterstützt werden. Mittels des Marketplace lassen sich jedoch diese Funktionen durch Community Lösungen ersetzen. Alternativ lassen sich auch Erweiterungen direkt im Filesystem auf dem Server einpflegen.

2.4 Dokumentation

Eine Dokumentation als Dokument wird in den Archiven nicht angeboten. Die Community Seite des BI-Servers bietet Informationen zur Versionierung, Aktualisierungen und Zugang zum Wiki[2] bzw. zum Forum[3]. Die Wiki Artikel sind größtenteils für die Version 2.x bis 3.x geschrieben.

2.5 Umsetzung Szenario

In diesem Kapitel werden alle Aktionen beschrieben, die zur Einrichtung der Arbeitsumgebung nötig sind. Zur Einrichtung der Arbeitsumgebung wurde zunächst eine 32-Bit Java Version installiert. Die Installation und Konfiguration von Pentaho teilt sich in den Bereichen Servertools und Clienttools.

[2] Siehe: http://wiki.pentaho.com/display/ServerDoc2x/BI+Server+2.x-3.x+Community+Documentation
[3] Siehe: http://community.pentaho.com/projects/bi_platform/

Zur Benutzung des MySQL Servers ist es notwendig, bei jedem Tool eine aktuelle Treiberdatei zu hinterlegen. Diese muss in den toolspezifischen Pfad abgelegt werden:

- **BI-Server:** /biserver-ce/tomcat/lib/
- **Admin Console:** /administration-console/jdbc/
- **Data Integration:** /data-integration/libext/JDBC/
- **Report Designer:** /report-designer/lib/jdbc/
- **Schema Workbench:** /pentaho/design-tools/schema-workbench/drivers/
- **Aggregation Designer:** /agg-designer/drivers/
- **Metadata Editor:** /metadata-editor/libext/JDBC/

BI-Server

Der Pentaho BI-Server nutzt als Infrastruktur einen Apache Server. Dieser muss zunächst als Dienst angemeldet werden. Dazu wird eine Eingabeaufforderung mit Administrator Rechten geöffnet und im Verzeichnis \biserver-ce\tomcat\bin die service.bat mit dem Parameter “install” ausgeführt. Anschließend die tomcat6w.exe als Administrator starten und in den Reitern Startup und Shutdown den kompletten Pfad zum Apache Tomcat Verzeichnis hinterlegen. Der Apache Tomcat Server lässt sich entweder im Dienstemanager oder mittels der tomcat6w.exe starten. Das Frontend des BI-Servers wird User Console genannt. Die User Console entspricht einem User Cockpit und kann mittels startpentaho.bat gestartet werden.

Admin Console

Die Admin Console ist das Monitoring Cockpit des Servers. Die Community Edition bietet nur wenige Funktionen. Es lassen sich lediglich Nutzer/-Rollen verwalten und wenige Serverfunktionen steuern (meist nur ein Refresh). Im Verzeichnis \biserver\administration-console\ lässt sich die Admin Console mittels start-pac.bat starten.

Die beiden Konsolen lassen sich mit einem Browser über folgende Daten erreichen:

User Console

url: http://localhost:8080/pentaho/

joe / password

Admin Console

url: http://localhost:8099/

admin / password

Clienttools

Eine Installation der Programme ist nicht nötig, sie liegen archiviert vor. In jedem Tool muss eine eigene Verbindung zur Datenbank mit folgenden Daten angelegt werden:

Hostname: localhost
Port: 3306
dbname: webshop
Driverclassname: com.mysql.jdbc.Driver
username: root
password: password
url: jdbc:mysql://localhost:3306/webshop

Zur Verwendung des OLAP Schema Workbench ist es notwendig im BI-Server in der Datei “publisher_config.xml” im Pfad “...\biserver\biserver-ce\pentaho-solutions\system” ein Publishing Passwort zu setzen (“password”).

Die Tools sind grundsätzlich alle in englischer Sprache vorliegend. Es gibt eine deutsche Übersetzung, diese ist aber nicht vollständig und wird daher hier nicht verwendet. Es ist möglich weitere Sprachpakete aus dem Community Forum einzubinden.

3 ETL-Prozess

3.1 Übersicht

Pentaho Data Integration ist das ETL Werkzeug von Pentaho. 2006 ging es aus dem Projekt Kettle hervor und basiert auf der Eclipse IDE. In PDI können Transformationen und Jobs modelliert werden. Als Job wird in PDI eine Orchestrierung von Transformationen beschrieben, aber auch weitere Jobs können in einem Job gesteuert werden. PDI verwendet ein Repository sowie ein Dateisystem zur Ablage der Projektdaten. Das Repository kann lokal oder auf einem Server angelegt werden. PDI speichert die ETL Pakete im .ktr Format, welches auf XML basiert.

3.2 Oberfläche und Bedienbarkeit

Nach Start des Tools[4] erwarten einen im Startbildschirm Hilfen und Beispiellösungen zum Einstieg in das Tool. Auf der linken Seite befinden sich die zwei Reiter View und Design. View listet alle im Repository verfügbaren Transformationen und Jobs auf. Der Design-Reiter enthält kategorisierte Modellierungsobjekte. Die Auswahlmöglichkeit der Modellierungsobjekte ist kontextabhängig (Transformation/Job).
Den Hauptteil der Oberfläche macht der Modellierungsbereich in der Mitte aus. In diesem werden per Drag&Drop Objekte aus dem Design-Tab verwendet. Beim Starten des Tools werden oberhalb des Modellierungsbereichs noch offene Modellierungen in Reitern organisiert angezeigt.
Oberhalb davon ist eine Auswahl anderer Perspektiven möglich. Die Modell und Visualize Perspektiven ermöglichen eine direkte Modellierung eines Metamodells zur grafischen Darstellung als Report oder Dashboard. Da dort Elemente der Business Edition zugänglich und die Benutzung dieser sehr instabil/fehlerhaft ist, ist anzunehmen, dass es nicht beabsichtigt ist diese in der Community Edition verfügbar zu haben.

[4] Server Dateipfad: C:\PDWH\pentaho\pdi-ce-4.4.0-stable\data-integration\Spoon.bat

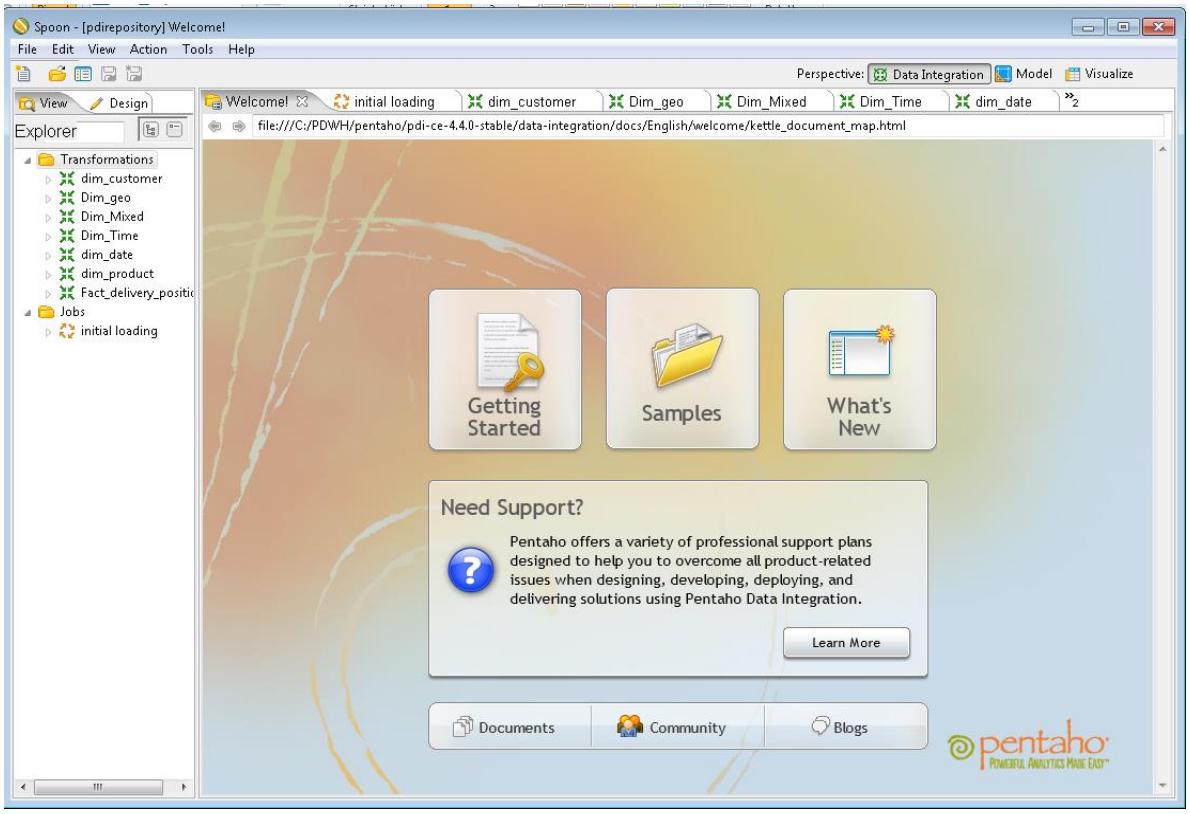


Abbildung 11 PDI Oberfläche

PDI lässt sich einfach und intuitiv mittels Drag&Drop und den kategorisierten Modellierungsobjekten bedienen. Neben den Standartobjekten der Modellierung bietet PDI viele vordefinierte Schnittstellen Plugins für externe Dienste/Werkzeuge. Zusätzlich lassen sich auch mittels Marketplace oder Eigenprogrammierung weitere Objekte einbinden.

3.3 Funktionalität

Wie eingangs bereits beschrieben werden mit PDI Transformationen modelliert und ausgeführt, sowie diese mittels Jobs orchestriert. In diesem Kapitel werden die wichtigsten Objekte zur Modellierung von Transformationen beschrieben. Diese wurden in die folgenden Kategorien unterteilt:

- Input,
- Manipulation,
- Flusssteuerung,
- Datenprüfung und Fehlerbehandlung,
- Datenbanken

Eine Übersicht über alle verfügbaren Objekte befindet sich im Anhang A5. Abschließend folgt eine Beschreibung der Job Funktionen.

3.3.1 Input

Ein Textinput mag in seiner Art zwar eine einfache Inputform sein, das Auslesen kann aber aufgrund seiner vielfältigen Konfigurationsmöglichkeiten aufwendiger sein. Pentaho Data Integration bietet dazu einige Unterstützung. Neben Textinputs ermöglicht PDI auch das Auslesen von Excel und XML Dateien(mittels XPATH) an. Des Weiteren lassen sich Ordner-/Dateistrukturen und Systemvariablen auslesen. Auf den Input mittels Datenbanken wird in Kapitel 3.3.5 eingegangen.

Für diese Inputmöglichkeiten gibt es die folgenden Modellierungsobjekte:

Tabelle 15 PDI Input-Modellierungsobjekte

Objekt	Beschreibung
Text file input	Textinputobjekt, ermöglicht auch mehrfaches Einlesen vieler Daten aus Ordnerstrukturen
Get data from XML	Ähnliche Struktur wie das Textinputobjekt, jedoch mit spezifischen XML Optionen.
Microsoft Excel Input	Beim Excelinput können mindestens eine Exceldatei und mindestens ein Excelsheet eingelesen werden. Sonstige Struktur ähnelt dem Textinput
Get System Info	Ermöglicht die Ausgabe von System-/Transformationsinformationen.

Der Aufbau der Datei-Inputobjekte ist grundlegend gleich, für ihre Spezialisierung aber angepasst. Der Textinput hat die allgemeinste Struktur und wird nachfolgend beschrieben.

Text File Input

Die Konfiguration ist in die Reiter File, Content, Error Handling, Filters, Fields und Additional Output Fields unterteilt.

Der Reiter File enthält die Konfiguration der Dateipfade, diese können entweder direkt angegeben werden, oder als Output eines vorherigen Objekts stammen. Zum Auslesen von Ordner- und Dateistrukturen werden die Objekte“Get File Names” und “Get SubFolder names” verwendet. So könnte beispielsweise eine Anwendung einen standardisierten Output in CSV Dateien in einen spezifischen Pfad ablegen, mit dynamischen Ordner-/Dateinamen. Diese dynamischen Order-/Dateinamen könnten mit diesen Objekten ausgelesen und an den Textinput übergeben werden. Alle Pfadangaben lassen sich mittels Regular Expression anpassen, wodurch auch dynamische Pfade als Regel erfasst werden können.

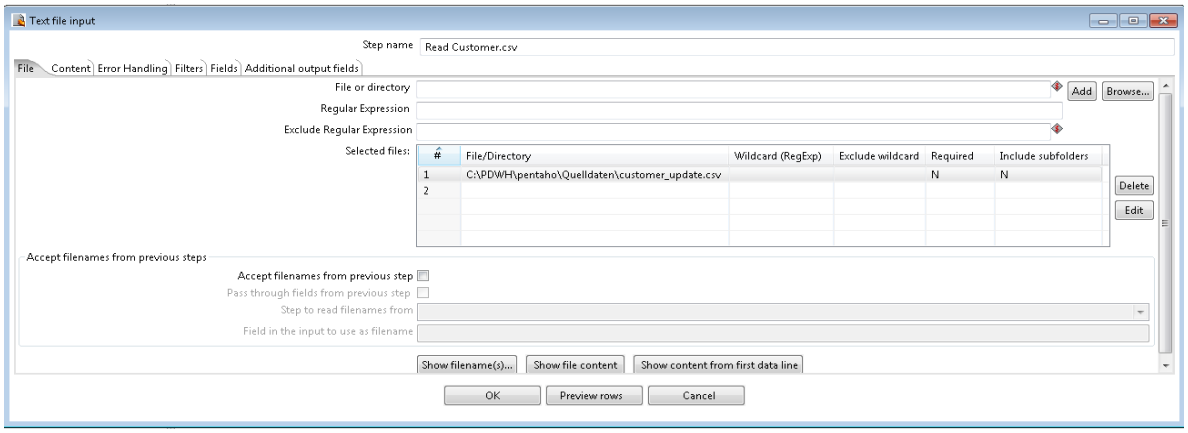


Abbildung 12 PDI Text File Input Objekt Übersicht

Der Reiter Content gliedert sich in die Bereiche Dateityp, Layout, Kompression und Format. Zunächst kann zwischen den Dateitypen “CSV” und “Fixed” gewählt werden, also Zeichentrennung oder feste Zeichenanzahl als Trennung. Dabei kann das Trennzeichen frei gewählt werden. Anschließend können Anzahl Header und Footer, Kompressionseinstellungen und Formateinstellungen definiert werden.

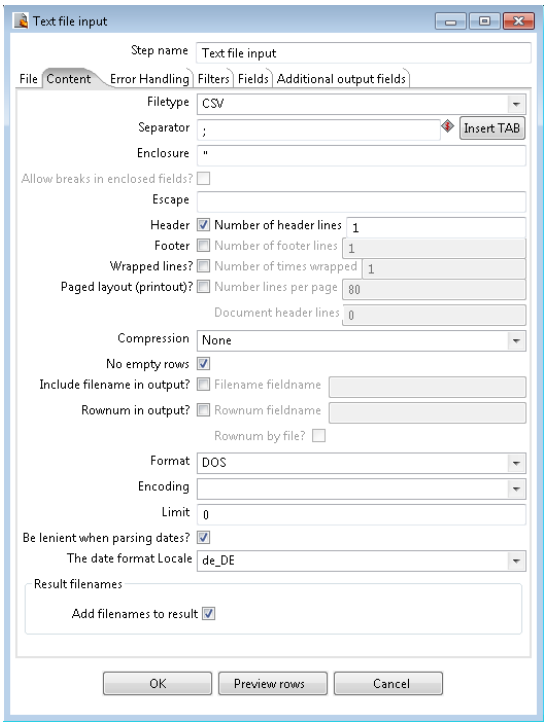


Abbildung 13 PDI Text File Input Reiter Content

Im Reiter Error Handling wird das Verhalten bei Auftreten eines Fehlers definiert. Es kann grundsätzlich in Ignorieren und Logging unterschieden werden. Das Logging entfernt die Zeilen vom Hauptfluss und legt diese in einer Datei ab. Zusätzlich fügt es den fehlerhaften Zeilen eine Fehlernummer, -feld, sowie -beschreibung hinzu. Order- und Dateistruktur lassen sich dabei entweder manuell anpassen oder aus einer Variablen auslesen.

Der Reiter Filters ermöglicht es nur spezifische Zeilen zu lesen oder diese auszulassen. Dabei wird die entweder die komplette Zeile oder nur eine spezifische Position nach einem Filterbegriff durchsucht.

Im Reiter Fields werden die Metadaten der Felder definiert. Dies kann entweder manuell erfolgen, oder mittels automatischen Auslesens.

Im Reiter Additional Output Fields können weitere Felder mit Metadaten der Datei erstellt werden, etwa Pfad, Größe und letzte Änderung.

Get System Data

Oft ist es nötig, zur Nachvollziehbarkeit von Änderungen oder Bearbeitungen Systeminformationen einer Zeile hinzuzufügen. Das “Get System Data” Objekt ermöglicht es, einige vordefinierte Informationen auszulesen und als Feld zu speichern.

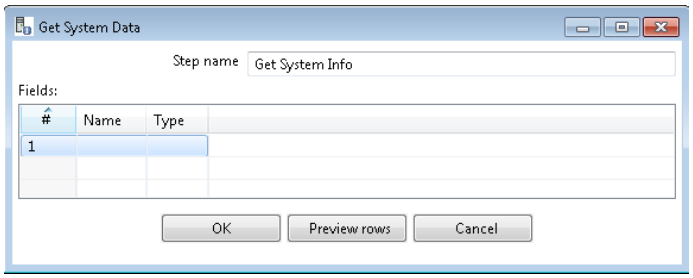


Abbildung 14 PDI Get System Data Objekt

Die vordefinierten Informationen können kategorisiert werden in Systeminformationen (Hardware, Systemzeit), allgemeine Zeit/Datumsfunktionen, Transformation- und Jobinformationen, sowie Informationen zur Java Virtual Machine.

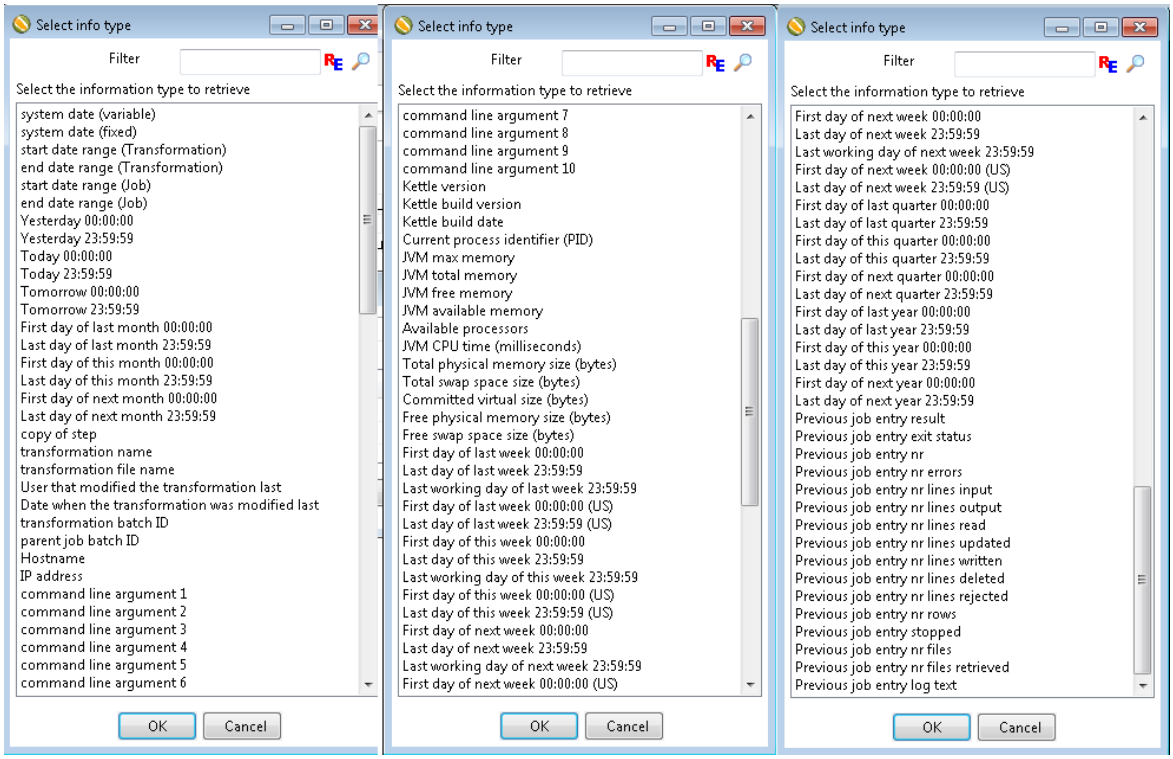

Select info type
Filter
Select the information type to retrieve
system date (variable)
system date (fixed)
start date range (Transformation)
end date range (Transformation)
start date range (Job)
end date range (Job)
Yesterday 00:00:00
Yesterday 23:59:59
Today 00:00:00
Today 23:59:59
Tomorrow 00:00:00
Tomorrow 23:59:59
First day of last month 00:00:00
Last day of last month 23:59:59
First day of this month 00:00:00
Last day of this month 23:59:59
First day of next month 00:00:00
Last day of next month 23:59:59
copy of step
transformation name
transformation file name
User that modified the transformation last
Date when the transformation was modified last
transformation batch ID
parent job batch ID
Hostname
IP address
command line argument 1
command line argument 2
command line argument 3
command line argument 4
command line argument 5
command line argument 6
OK
Cancel
Select info type
Filter
Select the information type to retrieve
command line argument 7
command line argument 8
command line argument 9
command line argument 10
Kettle version
Kettle build version
Kettle build date
Current process identifier (PID)
JVM max memory
JVM total memory
JVM free memory
JVM available memory
Available processors
JVM CPU time (milliseconds)
Total physical memory size (bytes)
Total swap space size (bytes)
Committed virtual size (bytes)
Free physical memory size (bytes)
Free swap space size (bytes)
First day of last week 00:00:00
Last day of last week 23:59:59
Last working day of last week 23:59:59
First day of last week 00:00:00 (US)
Last day of last week 23:59:59 (US)
First day of this week 00:00:00
Last day of this week 23:59:59
Last working day of this week 23:59:59
First day of this week 00:00:00 (US)
Last day of this week 23:59:59 (US)
First day of next week 00:00:00
Last day of next week 23:59:59
Last working day of next week 23:59:59
First day of next week 00:00:00 (US)
OK
Cancel
Select info type
Filter
Select the information type to retrieve
First day of next week 00:00:00
Last day of next week 23:59:59
Last working day of next week 23:59:59
First day of next week 00:00:00 (US)
Last day of next week 23:59:59 (US)
First day of last quarter 00:00:00
Last day of last quarter 23:59:59
First day of this quarter 00:00:00
Last day of this quarter 23:59:59
First day of next quarter 00:00:00
Last day of next quarter 23:59:59
First day of last year 00:00:00
Last day of last year 23:59:59
First day of this year 00:00:00
Last day of this year 23:59:59
First day of next year 00:00:00
Last day of next year 23:59:59
Previous job entry result
Previous job entry exit status
Previous job entry nr
Previous job entry nr errors
Previous job entry nr lines input
Previous job entry nr lines output
Previous job entry nr lines read
Previous job entry nr lines updated
Previous job entry nr lines written
Previous job entry nr lines deleted
Previous job entry nr lines rejected
Previous job entry nr rows
Previous job entry stopped
Previous job entry nr files
Previous job entry nr files retrieved
Previous job entry log text
OK
Cancel

Abbildung 15 PDI Auflistung Infotypes System Data Objekt

3.3.2 Manipulation

PDI bietet viele Möglichkeiten zum Hinzufügen, Verändern und Entfernen von Feldern. Dabei ist festzuhalten das es nicht den einen richtigen Weg oder das richtige Objekt für eine Tätigkeit gibt, sondern viele Objekte sich in ihrer Funktion überschneiden. Die Modellierungsobjekte lassen sich dafür prinzipiell in den Kategorien: Feldänderung, Sortieren und Filtern und Strukturänderung abgrenzen.

Feldänderung

Eine Feldänderung bedeutet, dass der Inhalt eines Felds erstellt oder verändert wurde. Die Erstellung von Inhalten kann dabei zwischen Generierung und Berechnung unterschieden werden.

Tabelle 16 PDI Manipulations-Modellierungsobjekte

Objekt	Beschreibung
Generierung	
Add constants	Definition von Feldmetadaten und konstanten Feldinhalt
Add sequence	Definition eines Counters, Intervallschritte, sowie Maximumwert
Berechnung	
Calculator	Berechnungen mittels fixer Formeln und maximal drei Feldern
Formula	freie Definition von Formeln, mittels fester Operatoren, aber freie Anzahl Felder
Group by	ermöglicht Aggregationen und Gruppierungen

Veränderung	
Value Mapper	Zuweisung bestehender Feldinhalt zu neuen Inhalt
String operations	Trimmen, entfernen von Tabstops/Special Characters, und Groß-/Kleinschreibung vereinheitlichen bei Feldinhaltstyp String
Replace in string	Ersetzen einzelner Zeichen bei Feldinhaltstyp String
Strings cut	schneidet einen String, Angabe von Startposition und Schnitt-länge

Zur Berechnung von Feldinhalten gibt es wie in Tabelle 16 angegeben die Möglichkeiten Calculator, Formular und Group by.

Das Calculator Objekt in PDI bietet nur sehr starre Möglichkeiten der Berechnung von Werten. Die Berechnung wird auf maximal drei Felder und feste Formeln beschränkt.

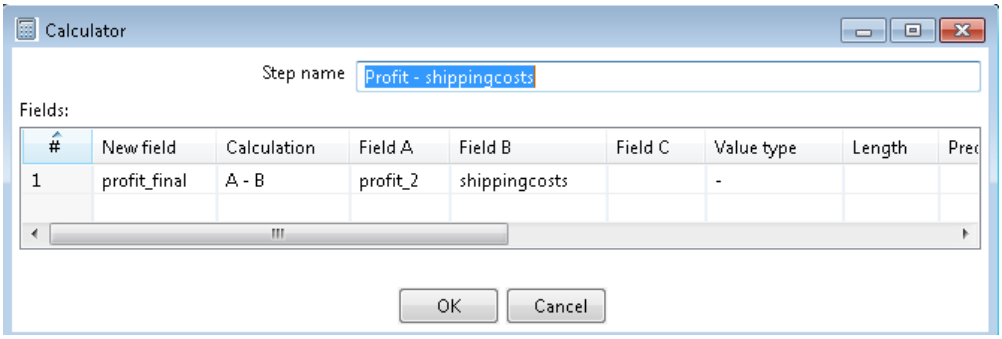


Abbildung 16 PDI Calculator Objekt

Die Auswahl der Formeln beschränkt sich dabei auf die Grundrechenarten, Datumsoperationen und String-Metriken.

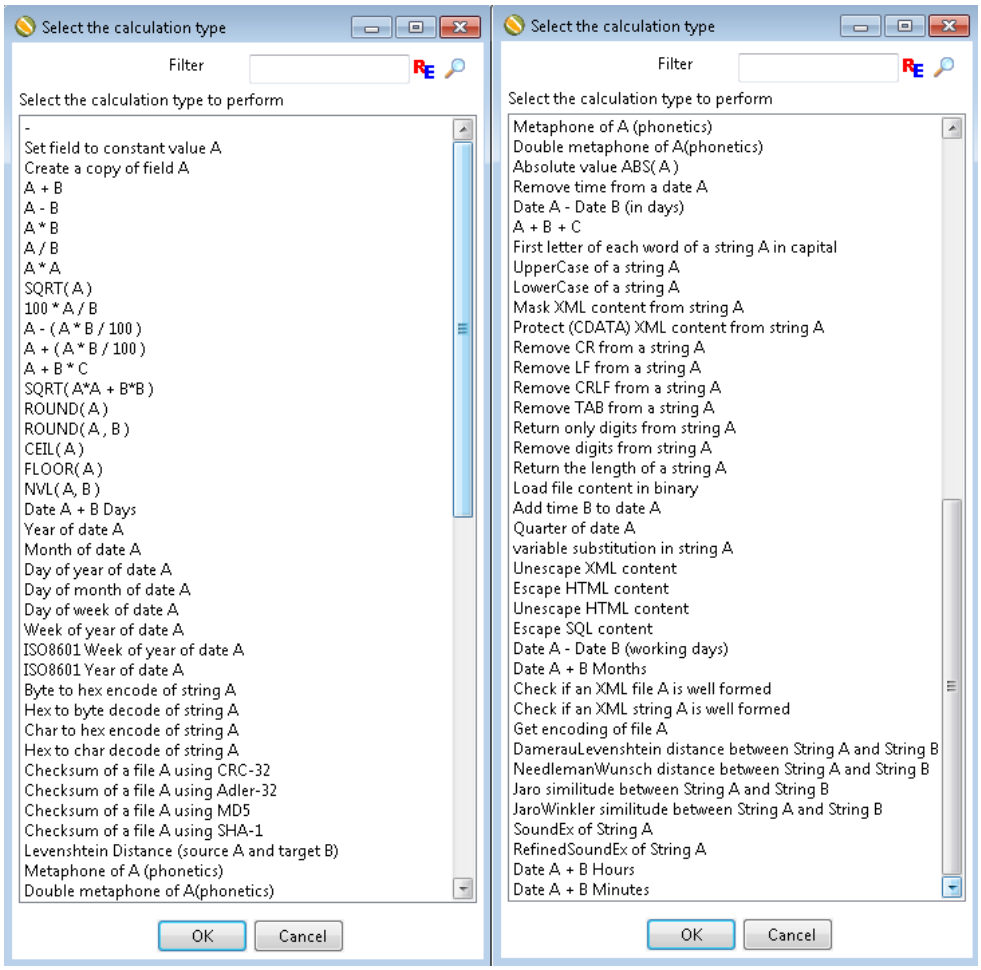

Select the calculation type
Filter
Select the calculation type to perform
-
Set field to constant value A
Create a copy of field A
A + B
A - B
A * B
A / B
A * A
SQRT(A)
100 * A / B
A - (A * B / 100)
A + (A * B / 100)
A + B * C
SQRT(A*A + B*B)
ROUND(A)
ROUND(A , B)
CEIL(A)
FLOOR(A)
NVL(A, B)
Date A + B Days
Year of date A
Month of date A
Day of year of date A
Day of month of date A
Day of week of date A
Week of year of date A
ISO8601 Week of year of date A
ISO8601 Year of date A
Byte to hex encode of string A
Hex to byte decode of string A
Char to hex encode of string A
Hex to char decode of string A
Checksum of a file A using CRC-32
Checksum of a file A using Adler-32
Checksum of a file A using MD5
Checksum of a file A using SHA-1
Levenshtein Distance (source A and target B)
Metaphone of A (phonetics)
Double metaphone of A(phonetics)
OK
Cancel
Select the calculation type
Filter
Select the calculation type to perform
Metaphone of A (phonetics)
Double metaphone of A(phonetics)
Absolute value ABS(A)
Remove time from a date A
Date A - Date B (in days)
A + B + C
First letter of each word of a string A in capital
UpperCase of a string A
LowerCase of a string A
Mask XML content from string A
Protect (CDATA) XML content from string A
Remove CR from a string A
Remove LF from a string A
Remove CRLF from a string A
Remove TAB from a string A
Return only digits from string A
Remove digits from string A
Return the length of a string A
Load file content in binary
Add time B to date A
Quarter of date A
variable substitution in string A
Unescape XML content
Escape HTML content
Unescape HTML content
Escape SQL content
Date A - Date B (working days)
Date A + B Months
Check if an XML file A is well formed
Check if an XML string A is well formed
Get encoding of file A
DamerauLevenshtein distance between String A and String B
NeedlemanWunsch distance between String A and String B
Jaro similitude between String A and String B
JaroWinkler similitude between String A and String B
SoundEx of String A
RefinedSoundEx of String A
Date A + B Hours
Date A + B Minutes
OK
Cancel

Abbildung 17 PDI Calculator Objekt Funktionen

Das Formular Objekt wird dagegen nur durch die Auswahl an Funktionen, jedoch nicht an der Anzahl Felder limitiert. Die Auswahl der Funktionen ist umfassend und beinhaltet auch logische Funktionen, sodass auch verschachtelte Formeln definierbar sind.

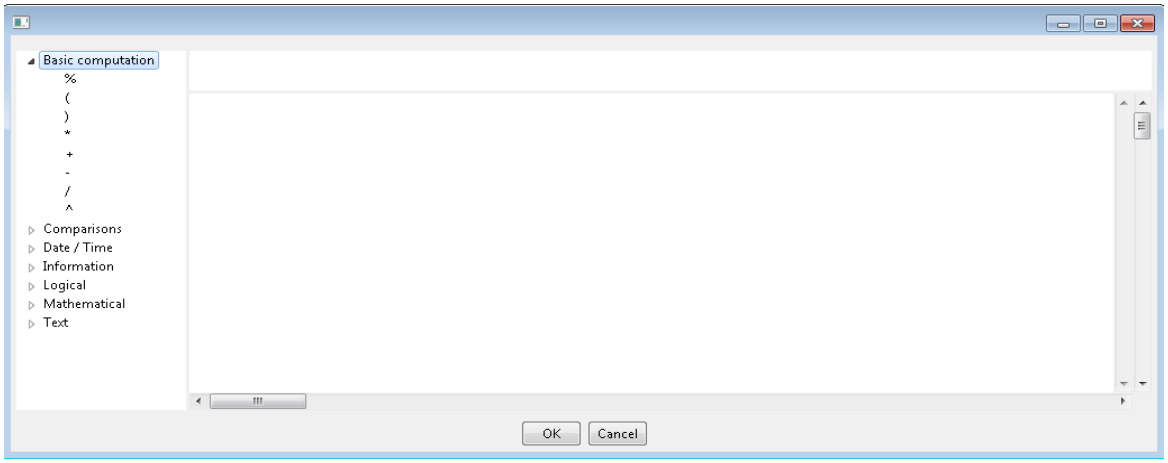


Abbildung 18 PDI Formular Objekt Funktionen

Die Existenz beider Objekte scheint durch Performanceunterschiede begründet zu sein.[5] Mittels Group By lassen sich die aus SQL bekannten Aggregationen und Gruppierungen auf den Datenstrom anwenden.

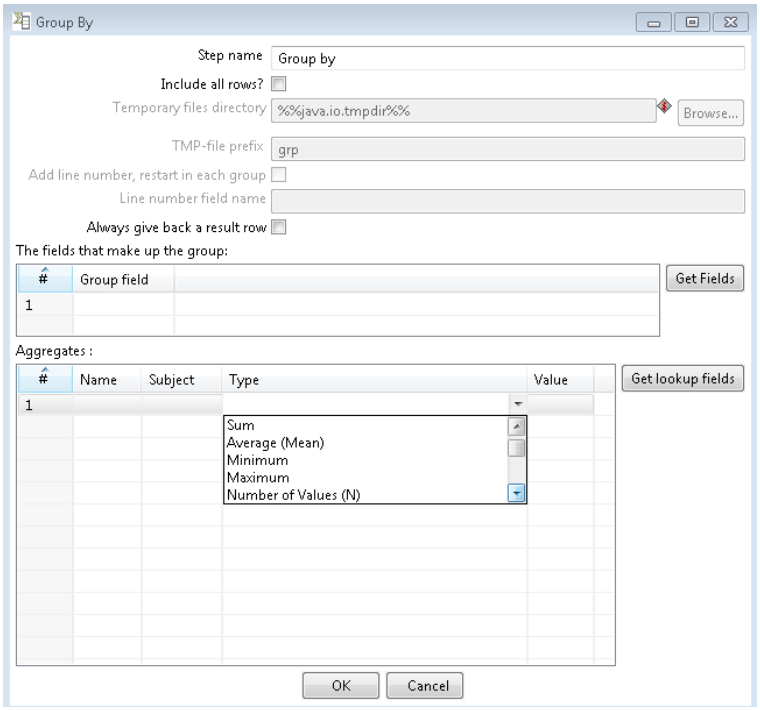


Abbildung 19 PDI Group By Objekt

Sortieren und Filtern

Zum Selektieren und Filtern von Zeilen/Spalten sind in Pentaho Data Integration vier Modellierungsobjekte vorhanden. Diese unterscheiden sich in den Möglichkeiten die Filterbedingungen und ausgehenden Datenflüsse zu konfigurieren.

Tabelle 17 PDI Filter/Sortierungs-Modellierungsobjekte

Objekt	Beschreibung
Select values	Selektieren, Entfernen und Metadatenänderung von Spalten

[5] Siehe: http://rpbouman.blogspot.de/2009/11/pentaho-data-integration-javascript.html

	Sortieren und Entfernen von Duplikaten (nur auf Schlüsselfeld), Zeilen basiert
	Filtert doppelte Zeileninhalte; nach definierten Spalten/alle Spalten; Trennung der Datenflüsse
	Zeilenbasierter Filter; voll konfigurierbarer Filter; Trennung der Datenflüsse nach Filterergebnis'true/false'

Mittels Select Values lassen sich Spalten aus einem Datenfluss entfernen/auswählen, sowie Metadaten der Spalten ändern. Somit kann der Datenstrom spaltenbasiert gefiltert werden.

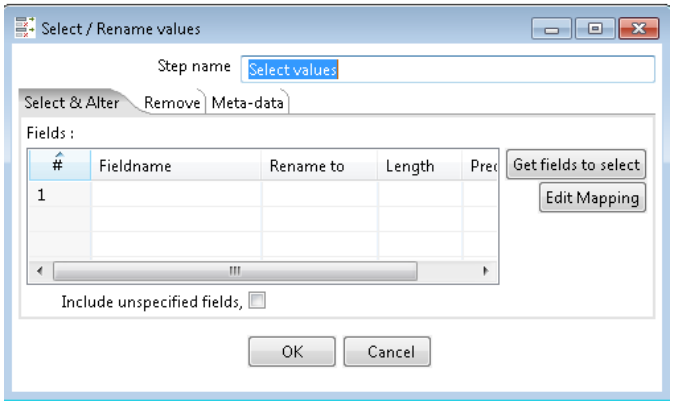


Abbildung 20 PDI Select Values Objekt

Das Sort Rows Objekt bietet die Möglichkeit, nach spezifischen Feldern den Datenstrom zu sortieren. Zusätzlich zur Sortierung können auch doppelte Zeilen entfernt werden. Dabei kann die Belastung der Maschine durch entweder fixe Zeilenanzahl pro Durchlauf oder freizuhaltenden Speicher(in Prozent) optimiert werden.

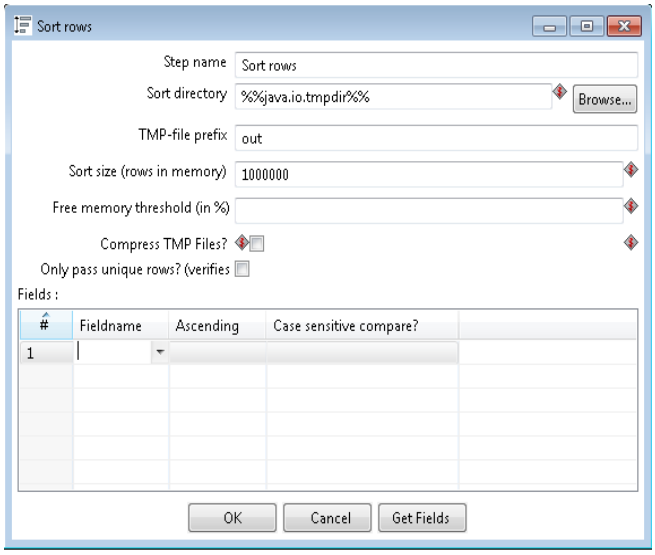


Abbildung 21 PDI Sort Rows Objekt

Zum Entfernen von Duplikaten kann auch das Objekt Unique Rows verwendet werden, jedoch müssen die Daten bereits sortiert sein. Der Vorteil dieses Objekts ist die Möglichkeit, die gefilterten Zeilen in einem zweiten ausgehenden Datenstrom separat zu behandeln.

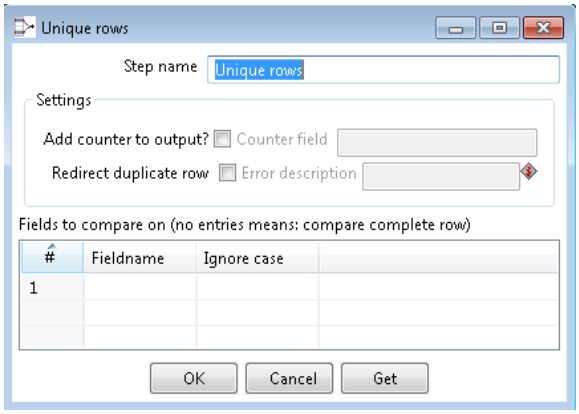


Abbildung 22 PDI Unique Rows Objekt

Mittels Filter Rows lassen sich Filter konfigurieren und Datenströme teilen. Es gehen bei diesem Objekt keine Zeilen verloren. Die Filterkondition kann dabei neben logischen Operatoren auch Regular Expression beinhalten.

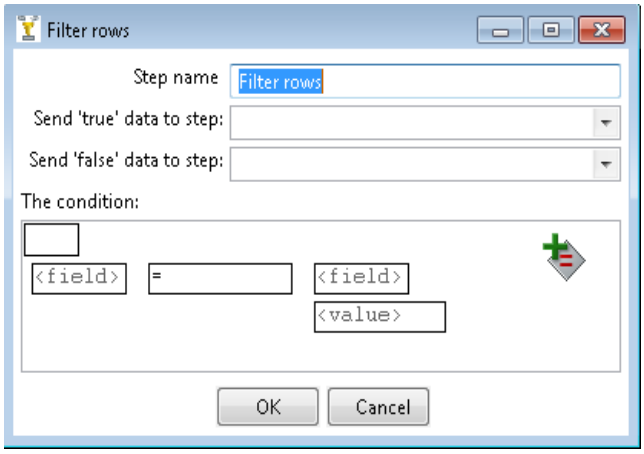


Abbildung 23 PDI Filter Rows Objekt

Strukturänderung

Pentaho Data Integration bietet die Möglichkeit Spalten zu trennen und vereinen, sowie Spalten zu Zeilen zu wandeln, sowie umgekehrt ((un-)pivot).

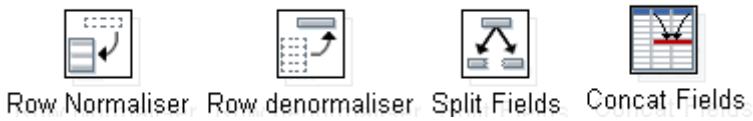


Abbildung 24 PDI Modellierungsobjekte zur Strukturänderung

Split Fields und Concat Fields bieten als Konfigurationsmöglichkeit jeweils nur das die Definition von Trennzeichen.

3.3.3 Flusssteuerung

Zum Trennen und Vereinen von Datenflüssen bietet Pentaho Data Integration verschiedene Konzepte, abhängig davon ob eine bestimmte Reihenfolge vorgesehen ist.

Das Trennen und Vereinen von Datenflüssen ist entweder mittels spezifischen Modellierungsobjekten oder mit einem direkten Eingriff in den Fluss möglich.

Tabelle 18 PDI Flusssteuerungs-Modellierungsobjekte

Objekt	Beschreibung
Prioritize streams	Vereinen vieler Flüsse, in definierter Reihenfolge
Sorted Merge	Vereinen vieler Flüsse, Reihenfolge durch Sortierungsfeld definiert
Merge Rows (diff)	Vereinen vieler Flüsse, löscht Duplikate, behält nur die neuere Zeile. legt Flag Field an mit Zustand der Zeile (deleted/new/identical/changed)
Merge Join	Vereinen von zwei Flüssen mittels Inner-/Leftouter-/Rightouter-/Fullouterjoin
Join Rows (cartesian product)	Vereinen von zwei Flüssen mit Crossjoin
Switch / Case	Trennen eines Flusses in n-Fälle

Zum Vereinen vieler Datenflüsse, ohne Sortierung, können alle Ausgangsflüsse in ein beliebiges Objekt geführt werden. Ein spezifisches Modellierungsobjekt ist nicht nötig.

Neben den Trennen und Vereinen von Datenflüssen aus fachlichen Gründen, lassen sich auch Flüsse zur Lastenteilung Trennen. In dem Kontextmenü lässt sich mit den Funktionen “Data movement” und “change number of copies to start” diese konfigurieren.

“Change number of copies to start” ermöglicht das Parallelisieren von Schritten. Dabei wird je Faktor eine weitere CPU mit der Verarbeitung belastet. Auf eine Datenbankaktion angewandt kann dies aber auch eine Performancesteigerung durch paralleler, offener Verbindungen hervorbringen.

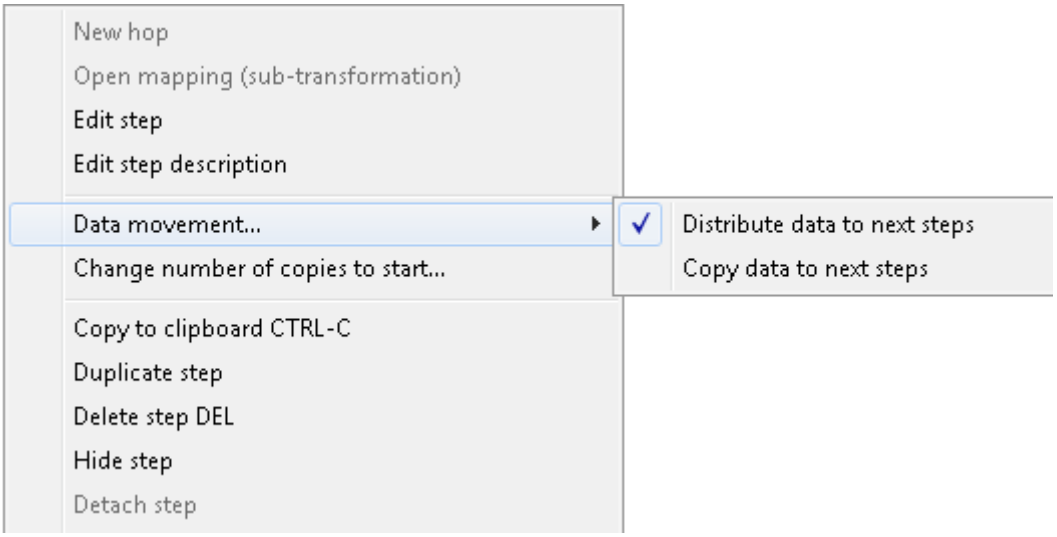


Abbildung 25 PDI Data Movement Typen

Mittels Data movement kann entschieden werden ob ein Datenfluss an mehrere Schritte kopiert (alle erhalten dieselbe Datenmenge) oder verteilt werden soll. Die Verteilung teilt dabei die Ausgangsdaten in n-Teile (n = Anzahl Schritte).

3.3.4 Datenprüfung und Fehlerbehandlung

Als Fehlerbehandlung wird im Folgenden die Unterstützung Pentahos bei Erkennen und Behandeln von Fehlern im Sinne der Datenqualität beschrieben. Die Reaktionen auf einen Fehler lassen sich auf die Vier Möglichkeiten beschränken:

1. keine Fehlerbehandlung und somit ein Weiterleiten des fehlerhaften Datensatzes
2. fehlerhaften Datensatz markieren und weiterleiten
3. fehlerhaften Datensatz zurückweisen
4. ETL Prozess stoppen

Ein Fehler im Sinne der Datenqualität bedeutet, ein Wert entspricht nicht dem erwarteten Wert. Etwa in Form von:

- nicht mögliche Werte (13.Monat etc.)
- Zahlen außerhalb der erwarteten Grenzen
- fehlerhafte Werte (fehlerhafter String / Spellcheck)
- Null Werte in Not Null Feldern (fehlende Werte)
- falscher Datentyp

Pentaho bietet zur Unterstützung das Data Validator Objekt. Dieses ermöglicht eine regelbasierte Fehlerbehandlung. Die Optionen sind in drei Teile gegliedert. Zunächst können die Fehlereigenschaften hinterlegt werden. Diese bestehen aus den vier Eigenschaften:

- Beschreibung (Fehlername)
- Inputfield
- Fehlercode (frei definierbarer Code)
- Fehlerbeschreibung (kurze Beschreibung des Fehlers)

Im “Type” Register lässt sich ein Datentypcheck konfigurieren, dabei lässt sich der erwartete Typ einstellen, sowie mögliche Trenn- und Gruppierungszeichendefinieren. Im “Data” Register wird eine Prüfung der konkreten Werte vollzogen. Die Werte können auf folgendes geprüft werden:

- Null Werte
- numeric Data
- Min/Max Stringlänge
- Min/Max Value
- erwarteter Suffix/Präfix (String)
- nicht erlaubter Suffix/Präfix (String)
- erlaubte Werte
    - entweder über das Kontextmenu eintragbar oder
    - aus eingehenden Schritt einlesbar
- Regular Expression Check

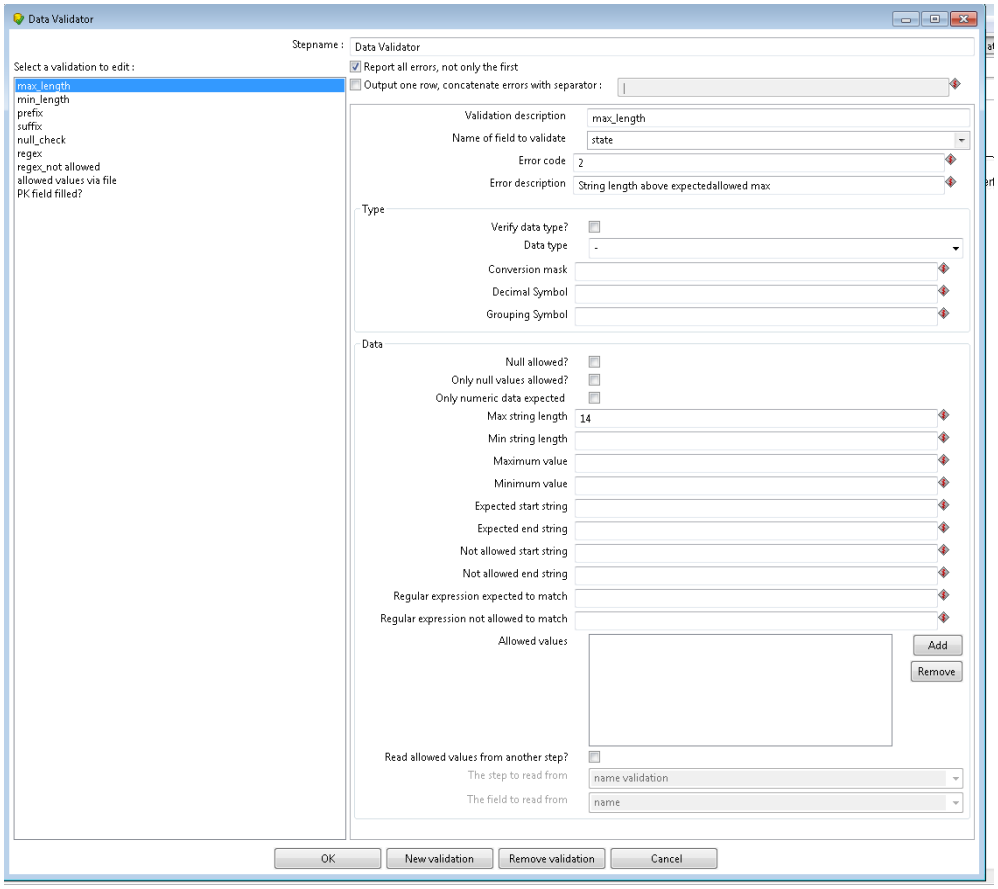


Abbildung 26 PDI Data Validator Objekt

Zur Nutzung von Regular Expression stellt Pentaho zusätzlich das Regex Evaluation Objekt zur Verfügung. Dieses ermöglicht Regular Expressions aufzustellen und an definierten Werten zu testen. Der Datenfluss wird in korrekte und fehlerhafte Daten getrennt. Für jeden Fehler und Datensatz wird eine Zeile im Fluss angelegt. Zusätzlich können die oben angelegten Spalten der Fehlereigenschaften mit eingefügt werden.

Neben der Fehlerbehandlung im Datenfluss bietet Pentaho Data Integration auch ein Logging auf Transaktionsebene. PDI unterstützt bei der Entwicklung von Transformationen mittels drei verschiedenen Loggings. Das Zeilenbasierte Logging kann in verschiedenen Detailstufen konfiguriert werden (von Error only, über Debugging, bis Rowlevel (feinster Grad)).

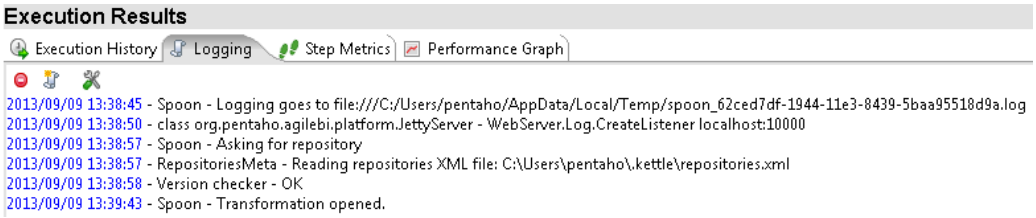
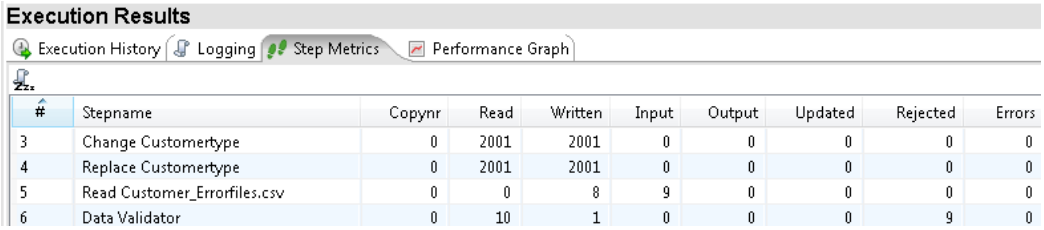

Execution Results
Execution History
Logging
Step Metrics
Performance Graph
2013/09/09 13:38:45 - Spoon - Logging goes to file:///C:/Users/pentaho/AppData/Local/Temp/spoon_62ced7df-1944-11e3-8439-5baa95518d9a.log
2013/09/09 13:38:50 - class org.pentaho.agilebi.platform.JettyServer - WebServer.Log.CreateListener localhost:10000
2013/09/09 13:38:57 - Spoon - Asking for repository
2013/09/09 13:38:57 - RepositoriesMeta - Reading repositories XML file: C:\Users\pentaho\.kettle\repositories.xml
2013/09/09 13:38:58 - Version checker - OK
2013/09/09 13:39:43 - Spoon - Transformation opened.

Abbildung 27 PDI Logging

Beim Step Metrics Logging werden für jeden Step die Anzahl Zeilen des Nachrichtenflusses geloggt. Dabei wird zwischen read, write, input, output, update, rejected und errors unterschieden.



#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors
3	Change Customertype	0	2001	2001	0	0	0	0	0
4	Replace Customertype	0	2001	2001	0	0	0	0	0
5	Read Customer_Errorfiles.csv	0	0	8	9	0	0	0	0
6	Data Validator	0	10	1	0	0	0	9	0

Abbildung 28 PDI Step Metrics Logging

Tritt ein Fehler auf wird dies im Logging und im Step Metrics angezeigt. Die Qualität der Fehlermeldungen ist sehr gut. Es wird präzise geloggt, bei welchem Step und zu welcher Uhrzeit ein Fehler geworfen wurde. Zusätzlich werden eine Fehlerbeschreibung und die Klasse des Fehlers angegeben.

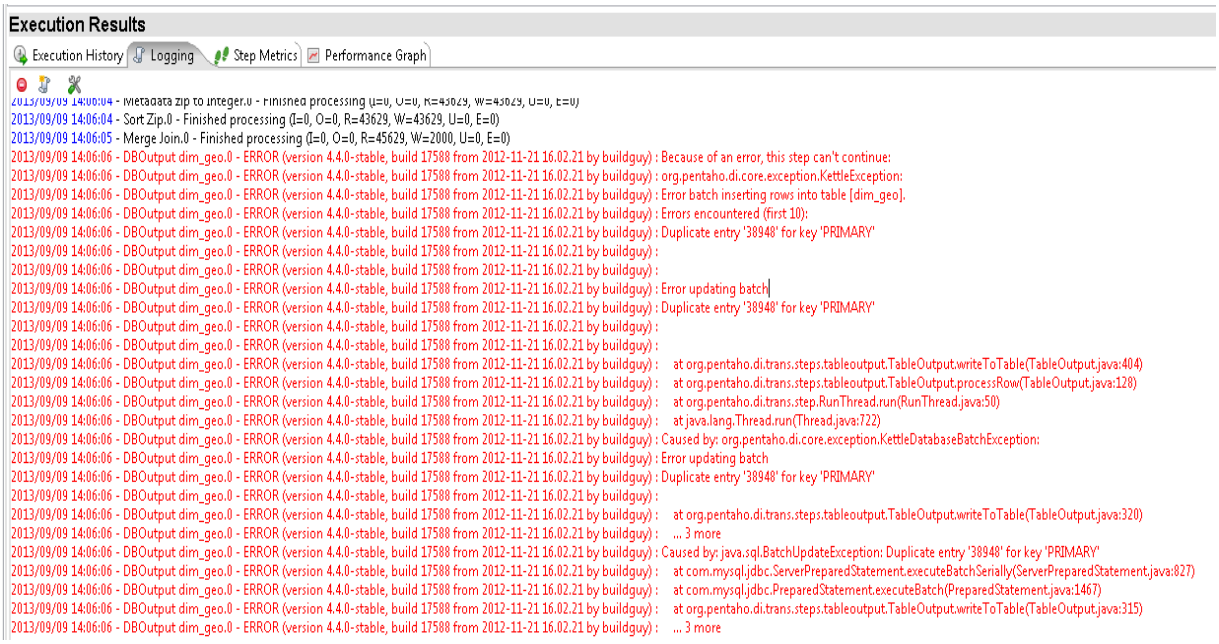


Abbildung 29 PDI Error Logging

Zur Visualisierung der Performance eignet sich neben den Step Metrics der Performance Graph. Dieser stellt die o. g. Metriken für einen ausgewählten Step grafisch dar.

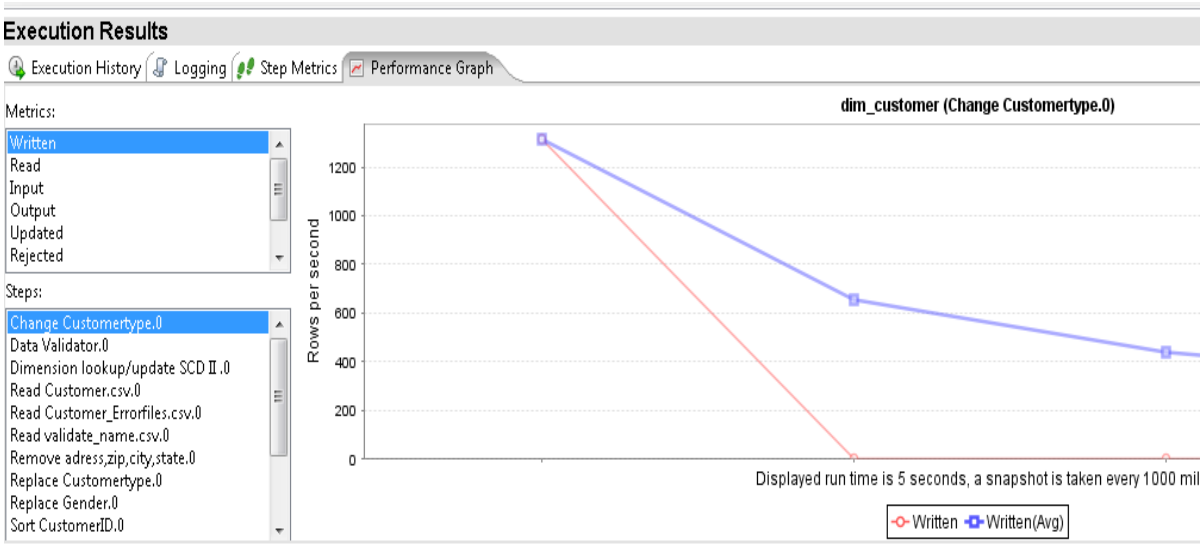


Abbildung 30 PDI Performance Logging

3.3.5 Datenbanken

Beim Arbeiten mit Datenbanken unterstützt Pentaho Data Integration mit vielen Assistenzfunktionen. Bei jedem Modellierungsobjekt im Datenbankkontext hat man die Möglichkeit die Datenbank zu durchsuchen und analysieren sowie eine Vorschau anzeigen zu lassen.

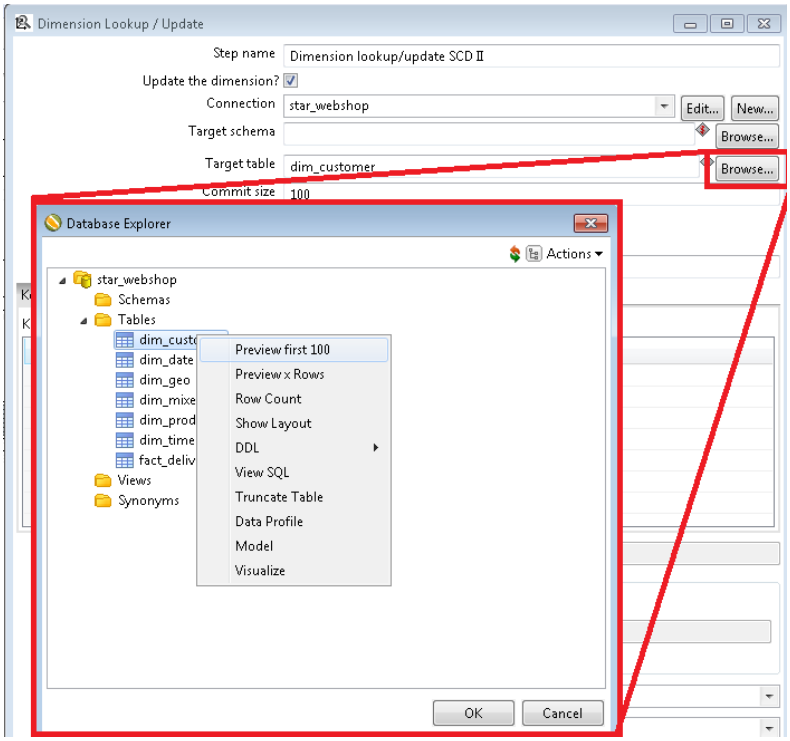


Abbildung 31 PDI Data Explorer

Die wichtigsten Modellierungsobjekte im Datenbankkontext sind Table input, Table output, Database lookup, Dimension lookup/update.

Tabelle 19 PDI Datenbank-Modellierungsobjekte

Objekt	Beschreibung
Table input	Auslesen einer Datenbank
Table output	Schreiben in eine Datenbank
Dimension lookup/update	Modellierungsobjekt zur Änderung von Slowly Changing Dimensions
Database lookup	Lookupobjekt

Table Input

Beim Table input führt der Assistent den Benutzer durch die Datenbank und generiert automatisch das SQL Statement, eine Anpassung ist dennoch möglich.

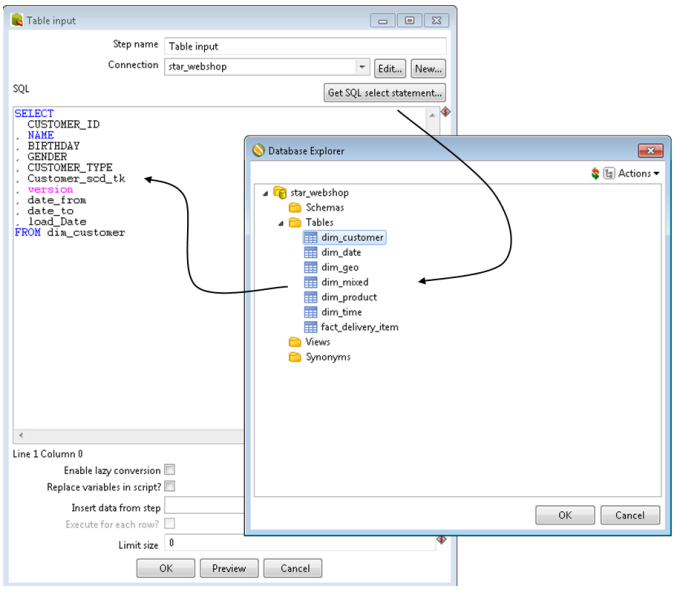


Abbildung 32 PDI Table Input Objekt SQL Assistenz

Table Output

Beim Table Output führt der Assistent ebenso den Benutzer zur Zieltabelle. Anschließend können die Spalten des Datenflusses mit den Tabellenspalten gemappt werden. Setzt man den Haken bei “specify Database Fields” nicht, so erfolgt ein automatisches Mapping, bei dem jedoch Quell- und Zielname identisch sein müssen. Des Weiteren ist es möglich vor jedem Insert die Tabelle mittels Truncate Table automatisch zu leeren.

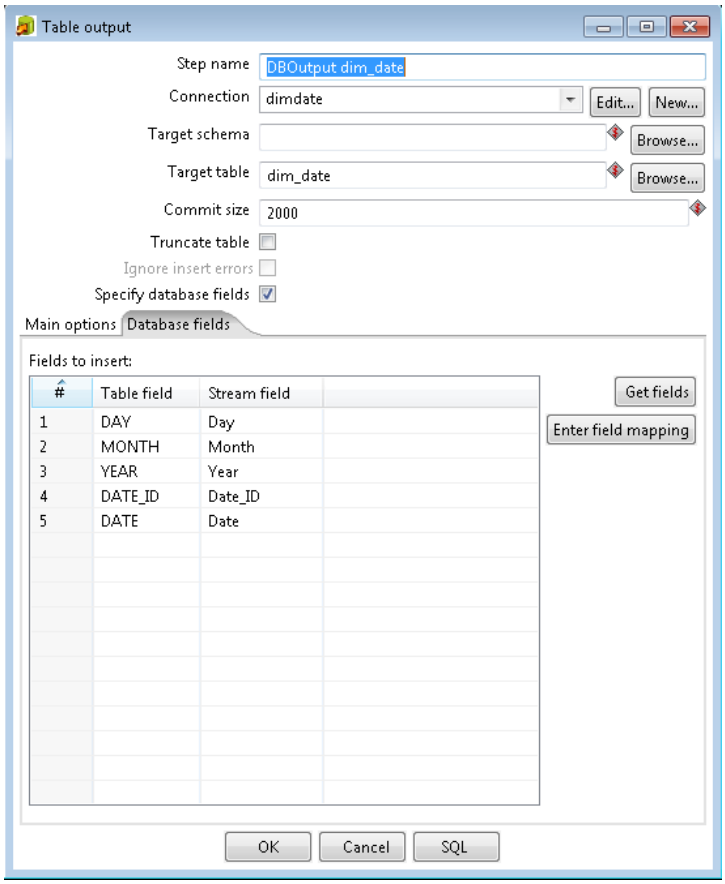


Abbildung 33 PDI Table Output Objekt

Dimension Lookup

Petanho ermöglicht eine einfache Implementierung des Slowly Changing Dimension Konzepts mittels dem Dimension Lookup / Update Objekts. Mittels des Assistenten kann, wie auch bei den vorherigen Objekten automisch die Zieldatenbank ausgewählt werden. Anschließend erfolgt die Auswahl des Schlüsselfelds und der Lookup Felder, dabei kann jedem Feld eine andere Aktion zugewiesen werden. Somit lassen sich auch Hybridtypen erstellen. Die Generierung des Werts des Technical Keys kann dabei entweder als autoincrement oder als Tablemax +1 erfolgen.

Für SCD Typ 2 lassen sich das gültige Start- und Enddatum beliebig festlegen.

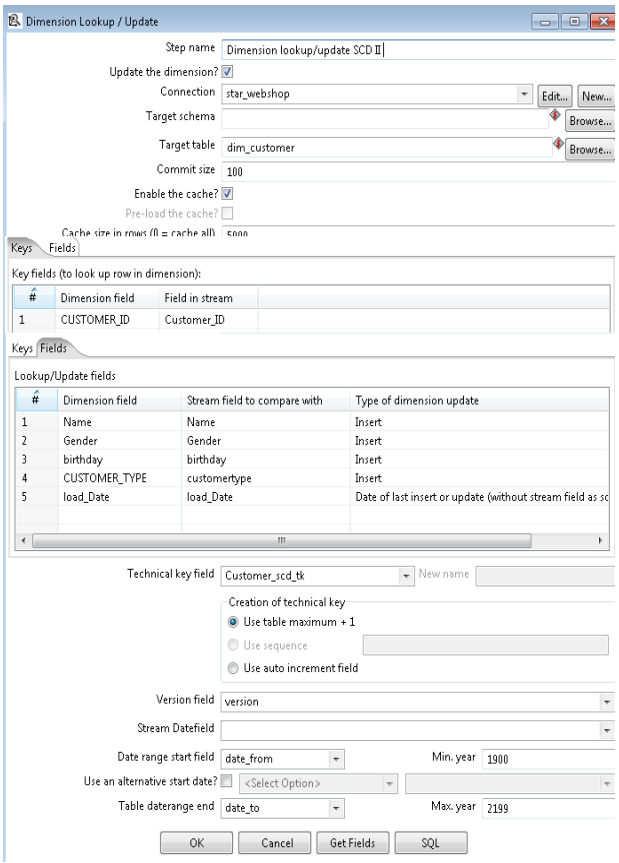


Abbildung 34 PDI Dimension Lookup / Update Objekt

Database Lookup

Beim Database Lookup Objekt wird zunächst die Lookuptabelle gewählt. Anschließend werden die zu durchsuchenden Tabellenspalten definiert und mit den Datenflussspalten gemappt. Abschließend wird die Spalte des Rückgabewerts wieder mit dem Datenfluss gemappt. Bei der Auswahl kann PDI einen über die Buttons “Get Fields” und “Get lookup Fields” Assistieren.

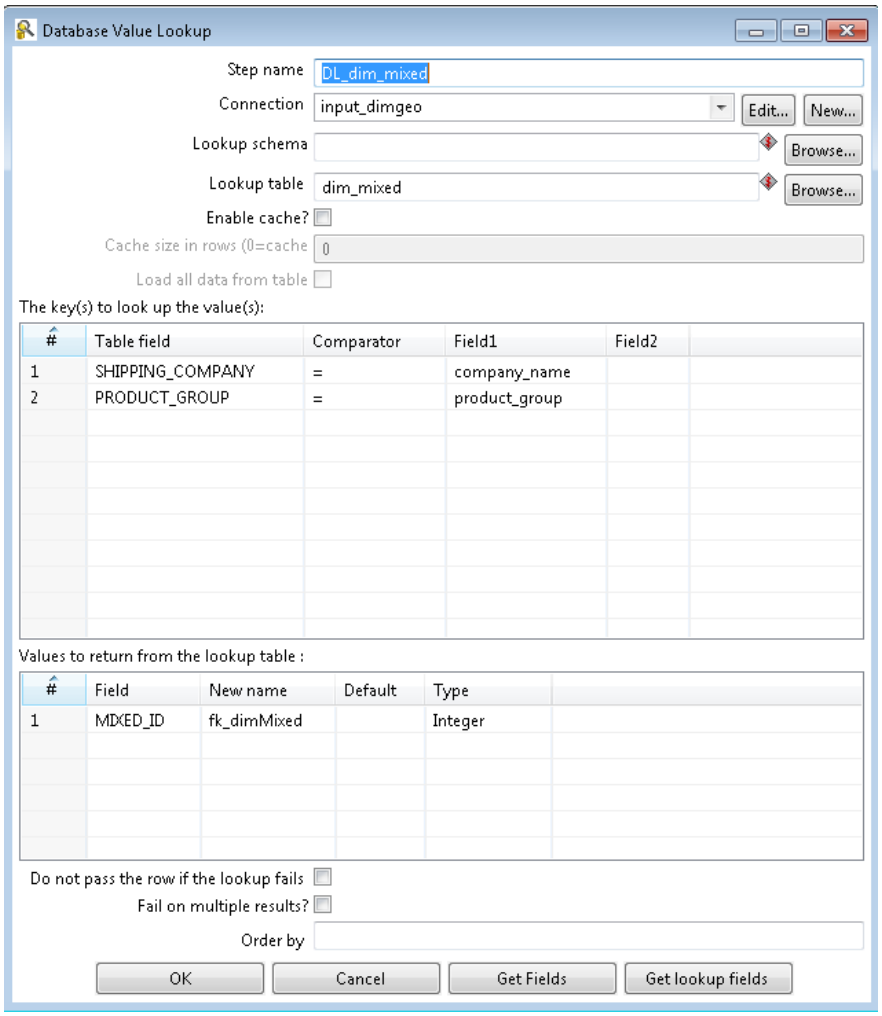


Abbildung 35 PDI Database Lookup Objekt

3.3.6 Job

Pentaho definiert einen Job als Organisationseinheit für Transformationen, vergleichbar mit dem Packet im Microsoft SSIS. Mit einem Job können zudem auch weitere Jobs gesteuert werden. Es besteht die Möglichkeit einen Zeitplan für einen Job einzustellen und diesen in definierbaren Perioden laufen zu lassen. In einem Job lassen sich diverse Aktionen zur Flusssteuerung und Automatisierung einbinden. Die folgende Tabelle stellt dafür kurz die wichtigsten Modellierungsobjekte vor.

Tabelle 20 PDI Job-Modellierungsobjekte

Modellierungsobjekt	Beschreibung
	Startobjekt des Jobs, beinhaltet Konfiguration des Zeitplans
	Endobjekt des Jobs
	Aufruf einer Transformation, Konfiguration der Lastverteilung über Cluster, Übergabe von Argumenten und Parameter in Transformation möglich
	Aufruf einer Jobs, Konfiguration der Lastverteilung über Cluster, Übergabe von Argumenten und Parameter in Transformation möglich
	Festlegen von Variablen, Gültigkeit dieser auf die Level JVM/current Job/parent Job/root Job einstellbar
	Ausführen einer .sql Datei oder Definition einer Query
	Ausführen von Javascript
	Prüft Status von Datenbankverbindungen
	Prüft Inhalt eines Ordners, Regex Wildcards nutzbar

	Prüft Status eines Webservices
	Erstellt eine Datei
	Verschiebt Dateien von/nach definierten Ordnern, Regex nutzbar
	löscht Dateien/Ordner, Regex nutzbar
	lädt Dateien über FTP, erfolgreiche Ausführung definierbar über Fehleranzahl/mindest Anzahl Dateien/vollständiger Erfolg
	speichert Dateien über FTP
	verschlüsselt Dateien mit PGP
	prüft Dateisignatur mit PGP
	entschlüsselt Dateien mit PGP
	Abbruchobjekt
	schreibt definierbaren Logeintrag

3.4 Dokumentation

PDI bietet bereits im Installationsarchiv diverse Hilfsangebote. Im Verzeichnis:...\pdi-ce-4.4.0-stable\data-integration\samples werden mehr als 200 Beispiele für Transformationen, Jobs und Mapping angeboten. Die Handbücher sind im Verzeichnis ...\pdi-ce-4.4.0-stable\data-integration\docs\English in diversen Sprachen verfügbar (nicht auf Deutsch). Diese sind jedoch nicht Community Edition spezifisch. Online sind im Infocenter (http://infocenter.pentaho.com/help/index.jsp) weitere Anleitungen und Hilfestellungen verfügbar.

Des Weiteren sind die Community und die Entwicklerblogs sehr aktiv und bieten einige Tutorials.[6] Im Pentaho Wiki sind zudem ausführliche Beschreibungen der im PDI verfügbaren Objekte.[7]

3.5 Umsetzung Szenario

Die in Kapitel 1.4.2 bereitgestellten Daten sollen nun extrahiert, transformiert und anschließend in das in Kapitel 1.4.3 beschriebene Starschema geladen werden. Zum Zweck des Funktionstests wurden nicht immer die idealen Schritte gewählt oder verschiedene Wege zur Lösung desselben Problems verwendet (insbesondere bei Textinputs und Stringoperationen).

3.5.1 Laden der Dimensionen

Dim_Customer

In die Dimension Dim_Customer werden die Kundendaten hinterlegt. Grundsätzlich werden die Daten aus der CSV Datei geladen und zunächst bereinigt. Das heißt in dem Fall das die nicht benötigten Informationen zu Adresse, Postleitzahl, Stadt und Bundesstaat entfernt werden. Anschließend werden die Genderstrings mit Kürzeln ersetzt. Anschließend wird der als Zahl vorliegende Kundentyp in ein String umgewandelt und durch einen Alias ersetzt (0 = private,1 = business). Abschließend wird mittels Dimension Lookup Objekt das Ergebnis in der Datenbank gespeichert.

[6] Wikipedia bietet eine gute Übersicht der Blogs:
https://en.wikipedia.org/wiki/Pentaho#Social_Media_Communication

[7] Siehe: http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+Steps)

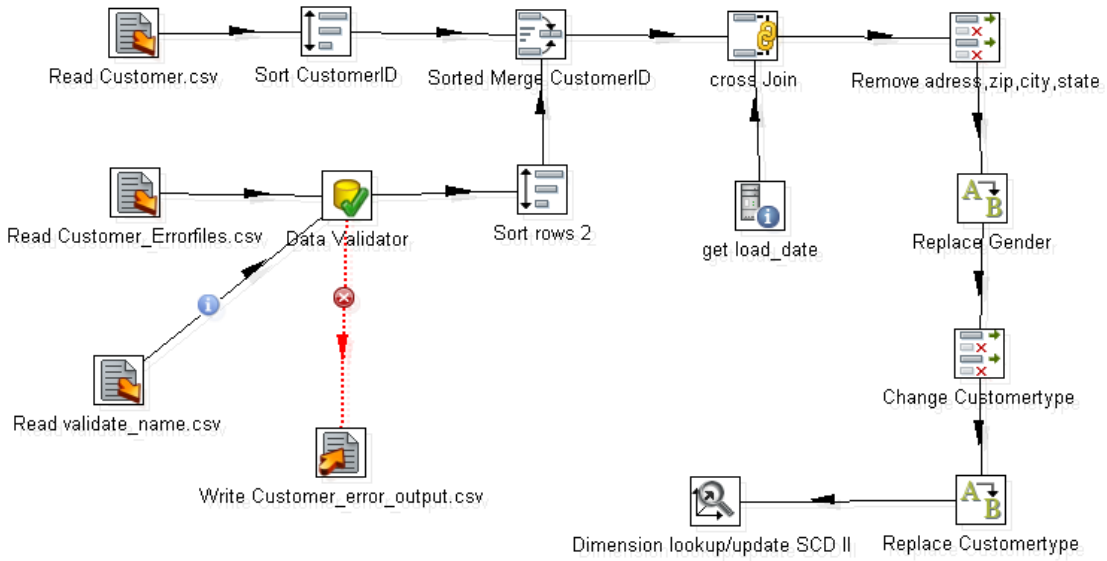


Abbildung 36 Transformation Dim_Customer

Vor dem durchführen des Dimension Lookup ist eine Anpassung an das Datenbankschema notwendig. Diese lässt sich mittels des Buttons “SQL” generieren (siehe Abbildung 32 PDI Table Input Objekt SQL Assistenz) und auf dem SQL Server ausführen. Hier wird als Lookupfield die CustomerID verwendet. Ist diese in den neuen Quelldaten enthalten und hat einen anderen Wert als die bisherige wird ein neuer Eintrag erstellt und mittels Insert befüllt. Der Primärschlüssel ist fortan “Customer_csd_tk”. Dieser wird automatisch befüllt mit dem Wert “table maximum +1”.

Dim_Geo

In diesem ETL Paket werden den in den Kundendaten hinterlegten Standortdaten die Latitude und Longitude hinzugefügt. Im Input “Read geolocation Textfile” wird die US.txt eingelesen. Es soll angenommen werden die vorliegenden Spalten haben kein eindeutiges Trennzeichen. Bereits im Input wird erkannt das der String in die Teile “Sonstiges” und “Latlon” getrennt werden kann. Innerhalb des “Latlon” Strings wird zunächst das Minus als Trennzeichen erkannt, entfernt und anschließend als Konstante wieder ein- und angefügt. Anschließend wird der Zipcode aus dem “Sonstige” String extrahiert und zu Integer konvertiert. Abschließend werden die Latitude und Longitude mit den verfügbaren Zipcodes aus der Customer.csv vereint und in der Tabelle Dim_geo abgelegt.

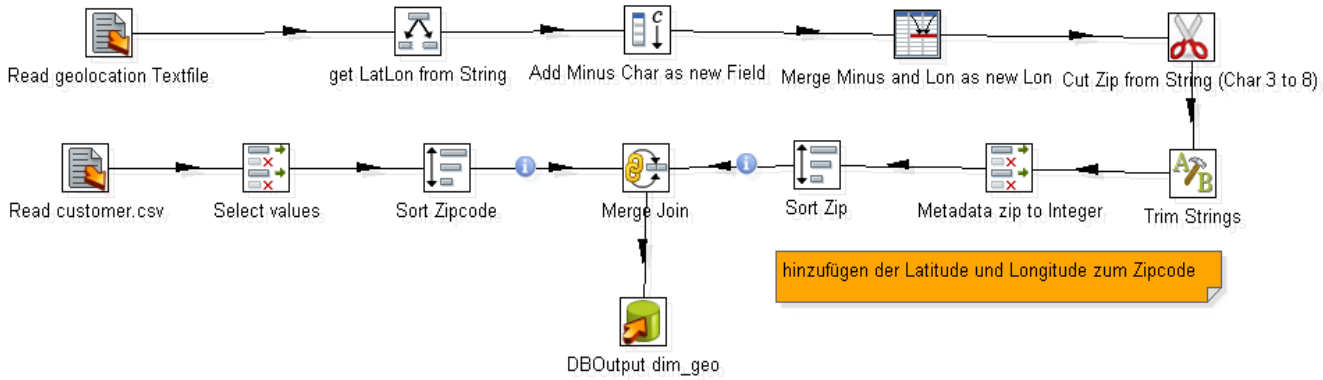


Abbildung 37 Transformation Dim_Geo

Dim_Mixed

Da nur sehr wenig Produktgruppen und Versandunternehmen vorliegen, sollen beide Informationen in einer Sammeldimension abgelegt werden. Zunächst werden die Produktgruppen aus der Product.csv extrahiert und Dopplungen entfernt. Anschließend werden Gruppenbezeichnern Werte zugeordnet (Zahl 1 bis 5). Diese fünf Werte werden mit den fünf Versandunternehmen mittels kartesischen Produkts zu 25 Einträgen. Abschließend wird mittels “Get Value from Sequence” eine ID hochgezählt. Diese dient als Eintrag für den Primärschlüssel.

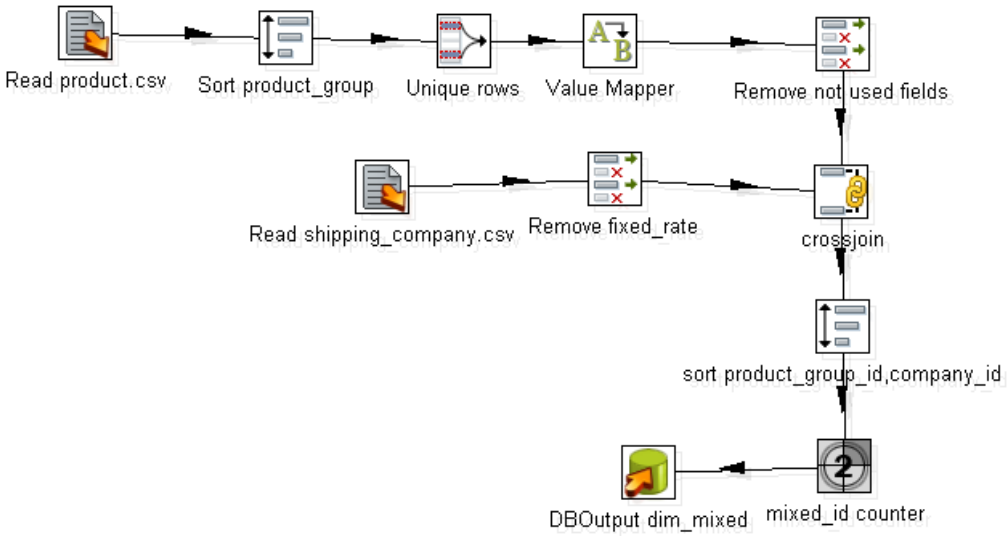


Abbildung 38 Transformation Dim_Mixed

Dim_Time

Die Dim_time soll sich aus bestehenden Daten aus der delivery.csv ableiten. Dazu wird das order_date ausgelesen und mittels Fieldsplitter Objekt in die Bestandteile Date und time gesplittet. Anschließend wird Time in hour/minute/second gesplittet und nach hour/minute sortiert. Zur Bildung der ID werden die Strings Hour und Minute mittels Calculator Object vereint.

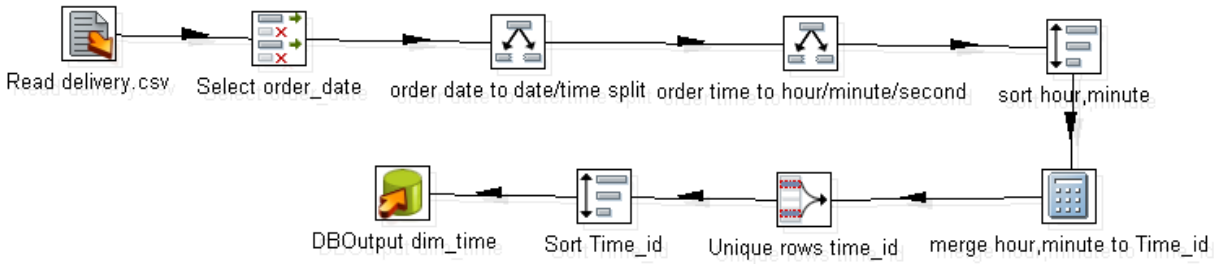


Abbildung 39 Transformation Dim_Time

Dim_Date

Die Dim_date verhält sich ähnlich zu der dim_time. Zur Erfassung aller Daten wird die delivery.csv zweimal eingelesen. Mittels Fieldsplitter werden die Strings delivery_date und order_date getrennt. Dabei wird jedoch nur vordere String weiterverwendet und in dem Feld Date hinterlegt. Anschließend werden die beiden Datenflüsse mittels Merge Rows (diff) vereint. Der Datestring wird gesplittet und anschließend in die Zielform transformiert.

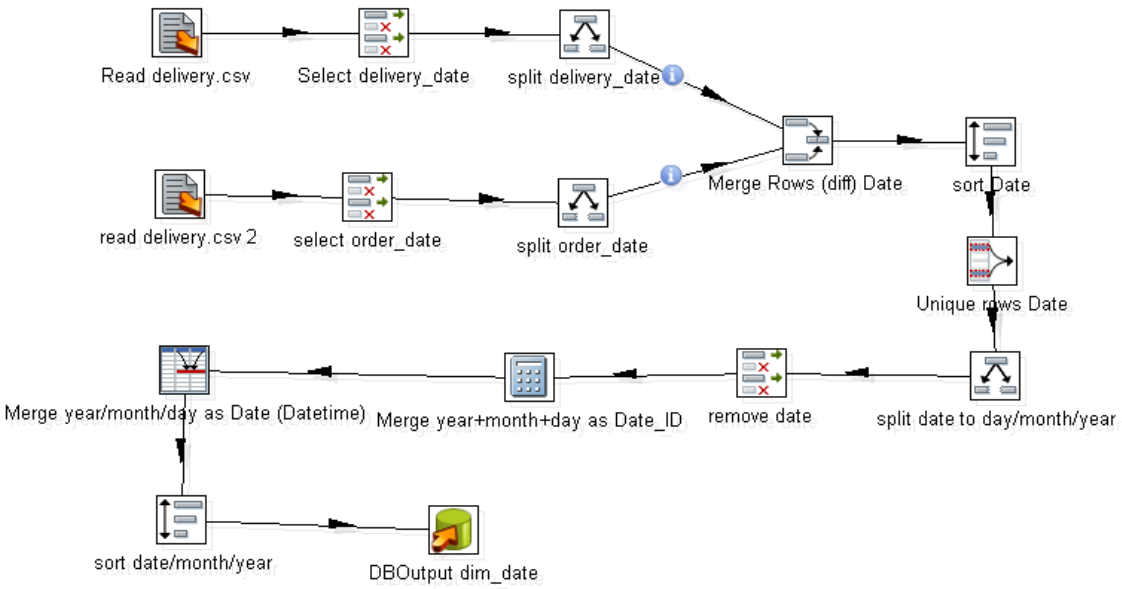


Abbildung 40 Transformation Dim_Date

Dim_Product

Bei der dim_product werden lediglich nicht benötigte Spalten entfernt und anschließend die Daten in der Datenbank gespeichert.

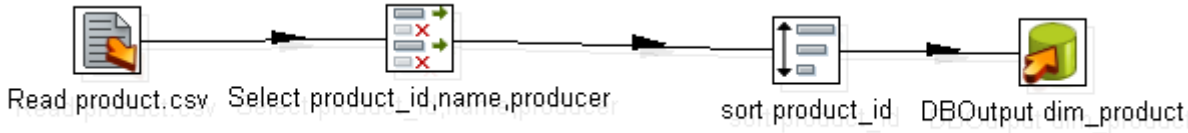


Abbildung 41 Transformation Dim_Product

3.5.2 Laden der Faktentabelle

Der ETL Prozess zur Befüllung der Faktentabelle ist in die drei Bereiche Extraktion der Quelldaten, Lookup auf Dimensionen und Berechnung der KPI und Laden in Zieltabelle gegliedert.

Die Quelldaten werden aus den .csv Dateien ausgelesen und mittels inner joins verknüpft.

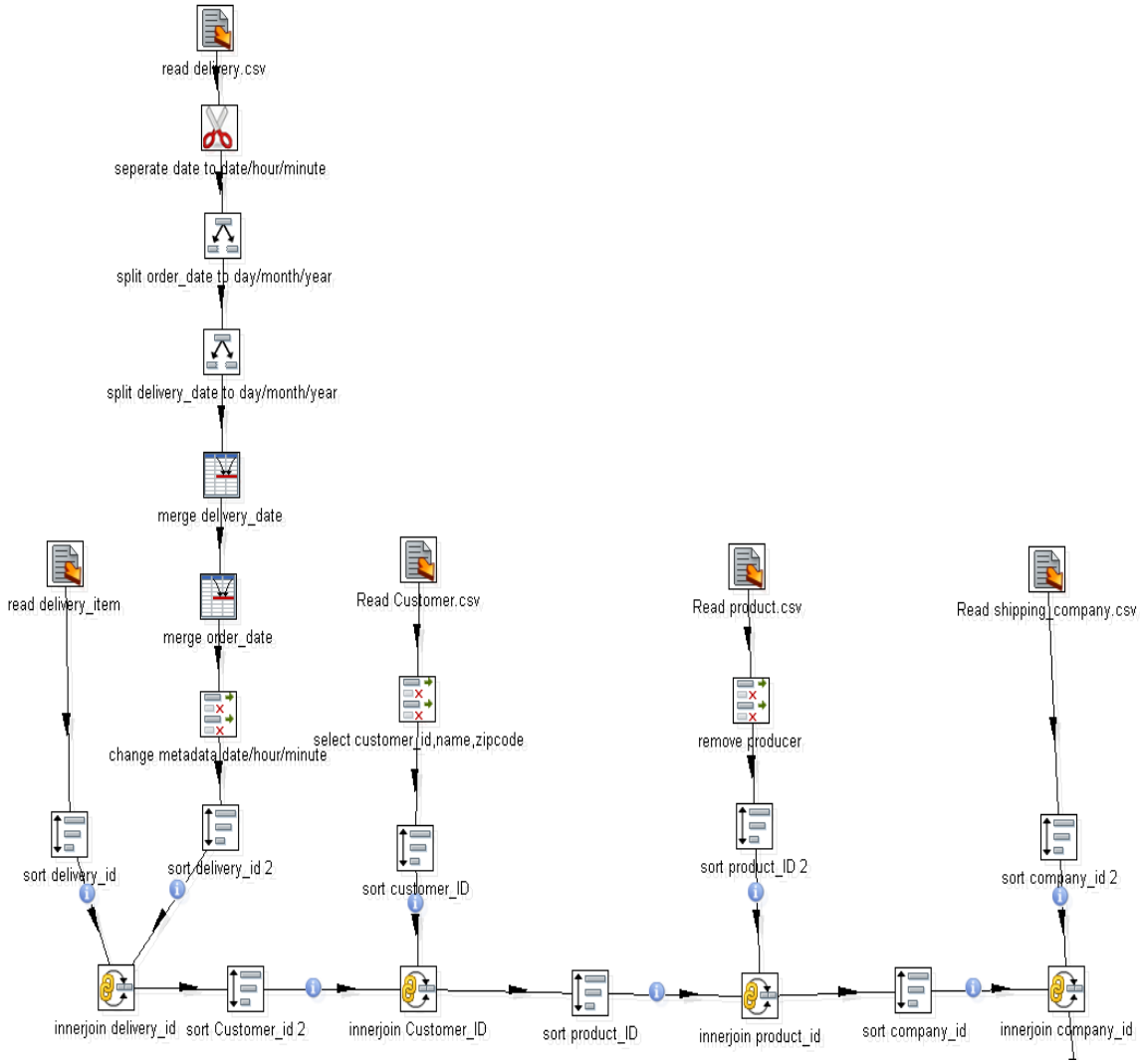


Abbildung 42 Transformation Fact_Delivery_item Extraktion

Anschließend werden Lookups auf die bestehenden Dimensionen durchgeführt um die zugehörigen ID's zu extrahieren.

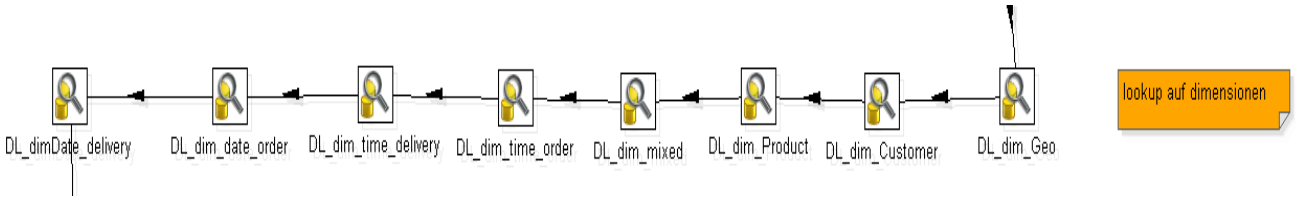


Abbildung 43 Transformation Fact_Delivery_item Lookups

Das Database Value Lookup Objekt ist sehr einfach zu konfigurieren. Zunächst wird die Connection und Lookuptabelle ausgewählt. Anschließend werden die Tabellenfelder definiert, auf die der Lookup durchgeführt werden soll und welchen Werten sie entsprechen sollen. Abschließend wird die Rückgabespalte der Tabelle ausgewählt.

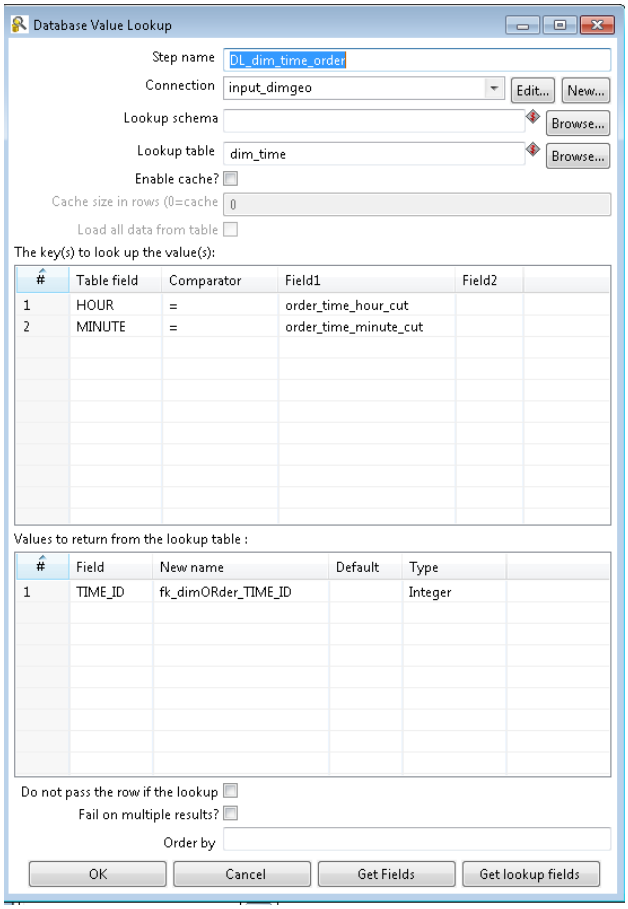


Abbildung 44 Transformation Fact_Delivery_item Lookup Dim_Time

Die Berechnung des Profits erfolgt mittels Calculator Objekten. Da im Calculator Objekt nur fest definierte Berechnungen mit maximal drei Feldern möglich sind, war es nötig die Berechnung in 6 Steps durchzuführen. Zunächst wird der Umsatz aus quantity und sales_price errechnet. Da es nicht direkt möglich war die Versandkosten auszugeben, müssen diese über den Faktor fixedrate zunächst auf den Umsatz aufgeschlagen werden. Anschließend wird der Umsatz davon abgezogen und übrig bleiben die Versandkosten. Zunächst wird der Profit pro Stück (noch ohne Versandkostenabschlag) ausgerechnet und anschließend mit der Positionsmenge multipliziert. Abschließend werden davon die Versandkosten abgezogen. Ergebnis ist der Profit pro Position.

- Turnover = quantity * sales_price
- Turnover_plus_shipping = turnover*(turnover*fixedrate/100)
- shippincosts=turnover_plus_shipping - turnover
- profit1 = sales_price - purchase_price
- profit2 = profit1*quantity
- profitfinal= profit2-shippincosts

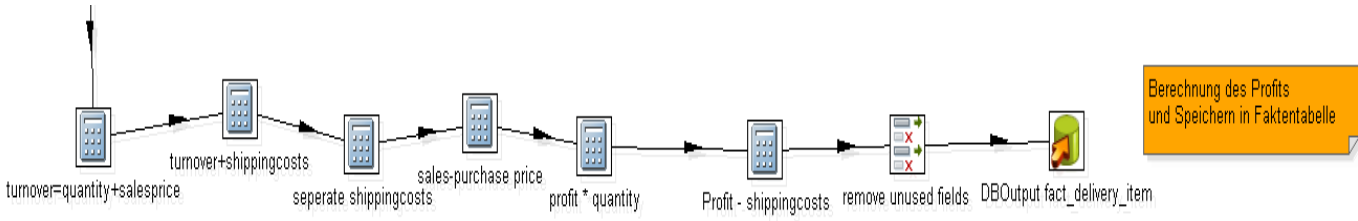


Abbildung 45 Transformation Fact_Delivery_item Kennzahlen und Speichern

Als Letztes werden die Daten in der Datenbank gespeichert. Um das Mapping beim Speichern in die Datenbank zu erleichtern, erfolgt zuvor eine Filterung der verfügbaren Felder.

4 OLAP

4.1 4.1 Übersicht

Pentaho Mondrian

Die technische Grundlage für multidimensionale Auswertungen stellt das OLAP-System “Pentaho Mondrian” dar. Als Bestandteil des BI-Servers ist diese standardmäßig in der Community Edition von Pentaho enthalten. Die in diesem Projekt verwendete Version ist 3.5.0. Aufgrund der Integration in den Server ist eine eigenständige Installation nicht vorgesehen. Pentaho Mondrian ist standardmäßig aktiv, nachdem der Server gestartet wurde. Die Architektur stellt sich wie folgt dar:

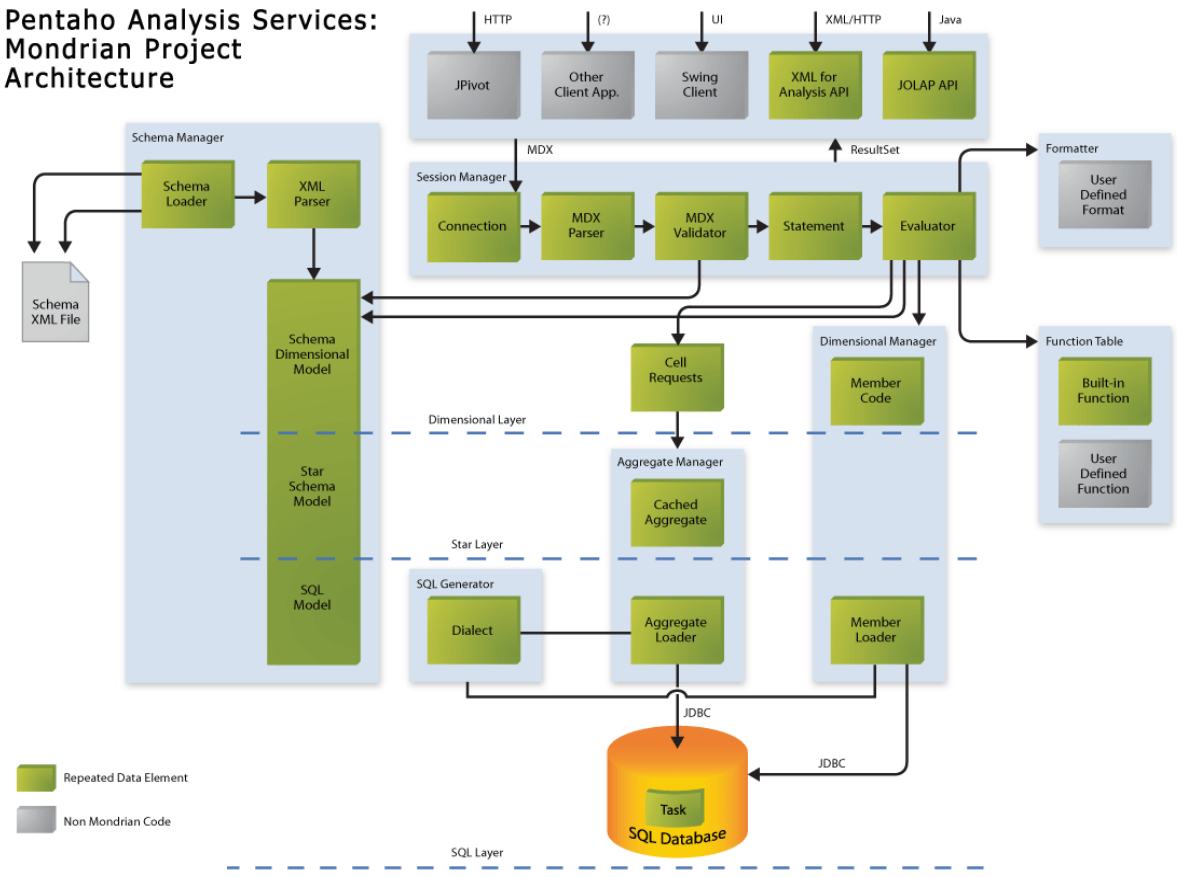


Abbildung 46 Architektur Pentaho Mondrian[8]

[8] Quelle: http://mondrian.pentaho.com/documentation/architecture.php

Das Pentaho Mondrian OLAP-System besteht aus vier Schichten, sog. Layers:

- **Presentation Layer:** Bestimmt was der Endnutzer auf seinem Monitor sieht. Die Darstellung ist von dem verwendeten Client-Tool abhängig. Der Client-Zugriff erfolgt mittels einer proprietären, auf Java basierenden API, über die MDX-Querys abgesetzt werden können. Der Server antwortet mit einem ResultSet.

- **Dimensional Layer:** Diese Schicht parst, validiert und führt MDX-Queries aus. Weiterhin werden dort die Metadaten des multidimensionalen OLAP-Schemas verarbeitet, auf welches im weiteren Verlauf näher eingegangen wird.

- **Star Layer:** Stellt Caching-Funktionalität bereit. Sollten Anfragen aus dem Dimensional Layer nicht aus dem Cache beantwortet werden können, wird eine neue Anfrage an den Storage Layer abgesendet.

- **Storage Layer:** Wird durch ein relationales Datenbankmanagementsystem (RDBMS) repräsentiert. Die MDX-Queries werden zunächst in SQL umgewandelt und dann an ein beliebiges RDBMS mittels JDBC abgesetzt. Pentaho Mondrian besitzt somit kein eigenes Storage-System und kann damit als ROLAP Server gesehen werden.[9]

Schema Workbench

Die Konfiguration und das Modellieren des den Abfragen zugrunde liegenden OLAP-Schemas wird primär über die sog. Schema Workbench durchgeführt. Die Schema Workbench, ebenfalls in Version 3.5.0 verfügbar, ist ein eigenständiges in Java geschriebenes Client-Tool, das als Modellierungs-Frontend fungiert und im Weiteren näher erläutert wird.

[9] vgl. http://mondrian.pentaho.com/documentation/architecture.php

4.2 Oberfläche und Bedienbarkeit

Nach Aufruf der Datei “workbench.bat”[10]öffnet sich das Hauptfenster der Schema Workbench, die nur in Englisch verfügbar ist. Es besteht aus einer Menüleiste mit den typischen Elementen “File”, “View”, “Options” usw. Von der Menüleiste aus lassen sich weitere Fenster innerhalb des Hauptfensters öffnen, wodurch mehrere Fenster nebeneinander dargestellt werden können.

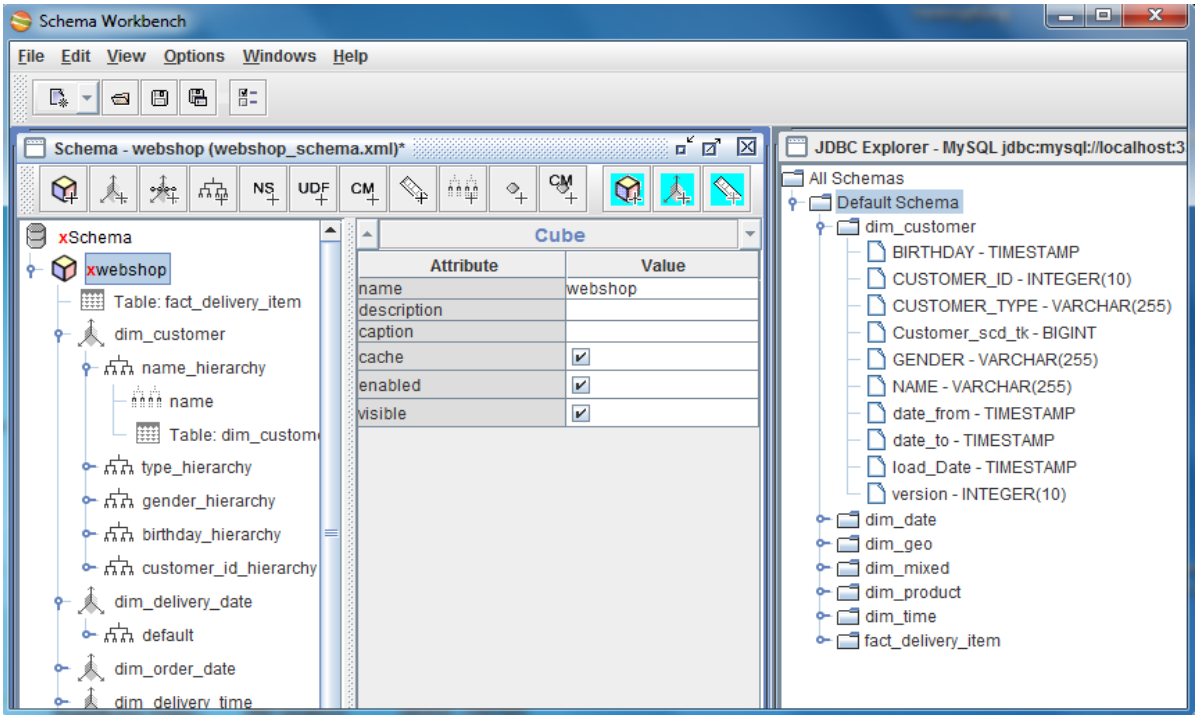


Abbildung 47 Übersicht Schema Workbench

Die Größe der Fenster sowie die Platzierung lässt sich individuell bestimmen. Bei einer Größenänderung kann es jedoch vorkommen das Bedienelemente “verschluckt” werden und erst wieder sichtbar sind, wenn die Fenster wieder vergrößert werden. Die Oberfläche macht insgesamt einen veralteten Eindruck. Im Gegensatz zu PDI basiert sie nicht auf dem Eclipse-Framework. Eine Aufteilung der Oberfläche mittels “Reiter” wie z. B. in PDI ist daher nicht vorhanden (vgl. Kapitel 3.2).

[10] Server Dateipfad: C:\PDWH\pentaho\psw-ce-3.5.0\schema-workbench\

Die Bedienung der Schema Workbench erfolgt mit Maus und Tastatur. Über die Symbolleiste, welche unterhalb der Menüleiste bzw. innerhalb jedes Fensters zur Verfügung steht, können die jeweiligen Hauptfunktionen aufgerufen werden. Weiterhin können diese Funktionen z. T. auch über das Kontextmenü (rechte Maustaste) aufgerufen werden. Die beim Erstellen eines neuen Schemas entstehende Baumstruktur enthält eine Ansammlung hierarchisch zueinander in Beziehung stehender Elemente (Tabellen, Kennzahlen, Dimensionen etc). Der Aufruf des Kontextmenüs auf den Elementen zeigt sämtliche für das jeweilige Objekt mögliche Operationen. Für jedes hinzugefügte Element können verschiedene Eigenschaften über tabellarisch dargestellte Felder vergeben werden.

4.3 Funktionalität

Die Pentaho Mondrian Engine führt MDX-Abfragen auf ROLAP-Schematas (Relational OLAP) aus. Ein solches Schema wird durch eine XML-Datei definiert, die in einer speziellen Struktur aufgebaut ist. Es kann als eine Cube-artige Struktur aufgefasst werden, die auf der Faktentabelle und den Dimensionstabellen im RDBMS basiert. Das Schema selbst besteht ausschließlich aus den in der XML-Datei definierten Metadaten. Es wird nicht separat als Cube in einer Datenbank abgespeichert sondern lediglich auf dem Pentaho BI-Server bzw. in Mondrian ausgeführt.

Die Schema Workbench stellt somit Funktionalität bereit, um ein Mondrian Schema erstellen zu können:

- Schema Editor: Modellieren, Validieren und Publizieren
- Absetzen und Testen von MDX-Queries gegen das erstellte Schema
- Anzeigen der Tabellen und Felder aus der relationalen Datenbank

4.3.1 Schema Editor

Um mit der Schema Workbench arbeiten zu können muss zunächst eine Datenbankverbindung zur relationalen Datenbank aufgebaut werden, welche die zugrunde liegenden Tabellen in Form eines Star-Schematas enthalten. Unter “Options → Connection” können dazu eine Vielzahl von Verbindungstypen ausgewählt und die Verbindungsdaten eingegeben werden.

Ein neues Schema kann über “File → New → Schema” angelegt werden. Es besteht aus einem Cube in der obersten Hierarchiestufe, der Faktentabelle, den Dimensionen inkl. Hierarchien, sowie den Kennzahlen (=Measures). Alle genannten Elemente müssen dem Schema einzeln hinzugefügt und als attribute–value pair in einer Tabelle konfiguriert werden. Dabei wird u. a. ein Mapping auf die Tabellen und Schlüsselfelder des relationalen Schemas durchgeführt wird. Eine automatische Vervollständigung von Feldnamen oder eine Einschränkung auf mögliche Felder gibt es nicht. Die genaue Kenntnis des relationalen Modells bzw. der Feldnamen wird daher vorausgesetzt. Die Eingaben werden nach Verlassen des Eingabefeldes sofort auf Kompatibilität mit der Mondrian Engine geprüft bzw. validiert. Sollten die hinzugefügten Elemente nicht richtig oder unvollständig konfiguriert sein, wird dies mit einem roten “x” beim fehlerhaften Element und einer in rot gehaltenen Fehlermeldung am unteren Fensterrand angezeigt.

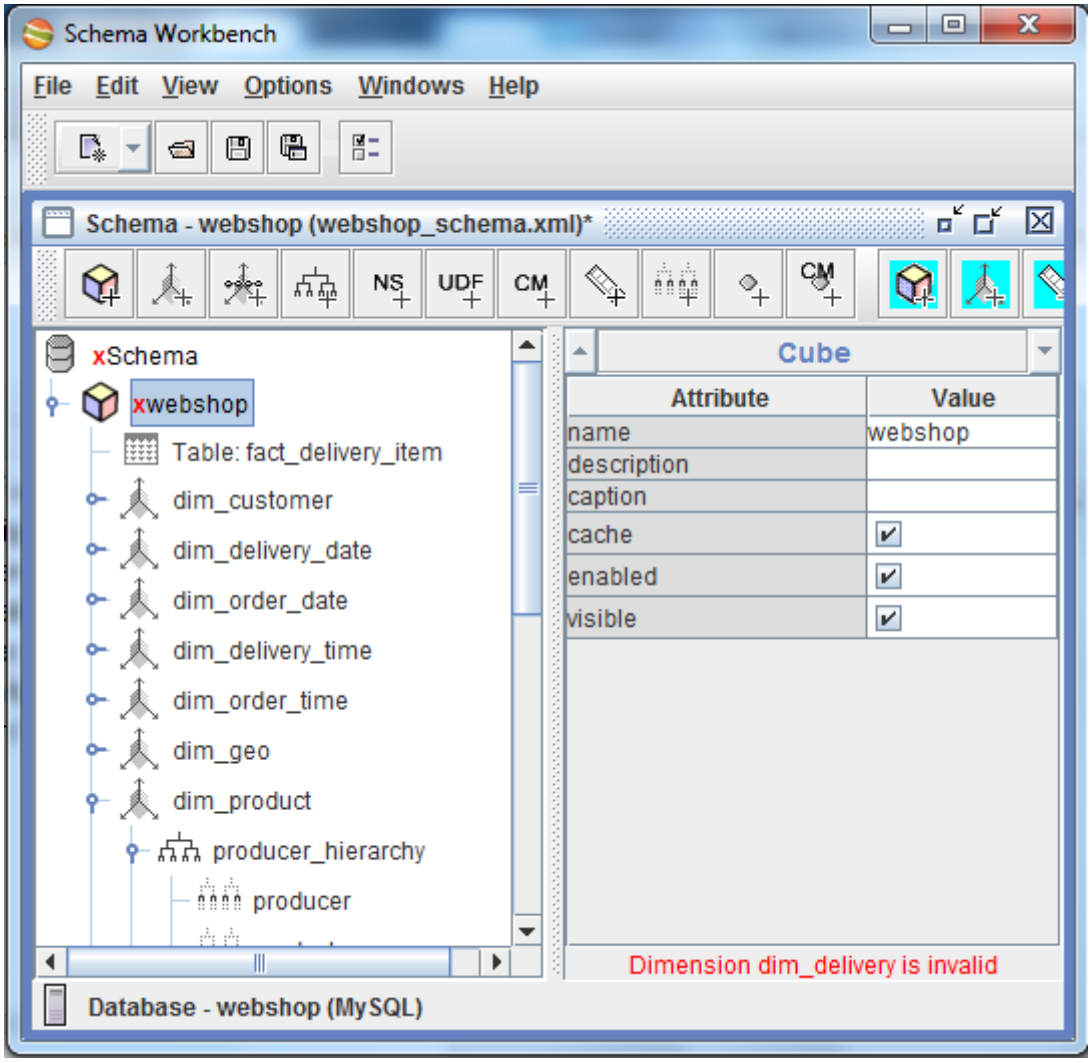


Abbildung 48 PSW Übersicht Schema Editor

Ein fertig modelliertes Mondrian Schema kann mittels “File → Save as” als XML-Datei im Filesystem (an beliebiger Stelle) gespeichert werden.

4.3.2 MDX-Queries

Bereits während der Modellierung und unabhängig vom Deployment auf dem BI-Server können aus dem Schema Modeller MDX-Queries auf das Schema abgesetzt und das Ergebnis angezeigt werden. Eine entsprechende Eingabemaske kann über “File → New → MDX Query” aufgerufen werden. Es werden jedoch keinerlei Eingabehilfen unterstützt. Die Ergebnisausgabe ist nicht tabellarisch sondern an eine spezielle Form der Konsolenausgabe angelehnt.

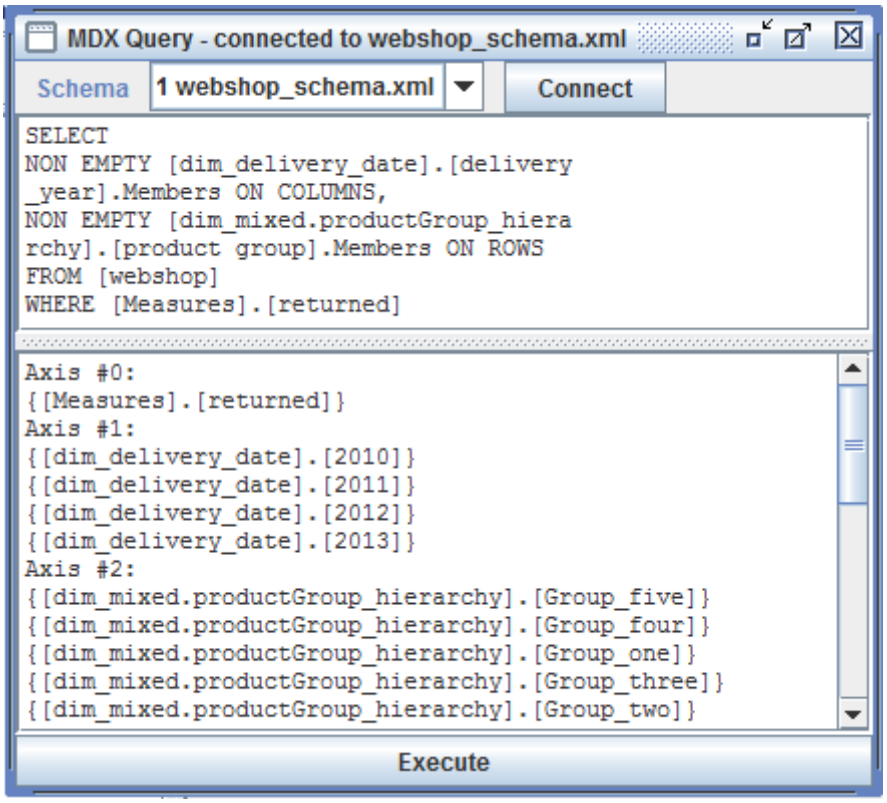


Abbildung 49 PSW Beispiel MDX-Query

Die Möglichkeit, MDX-Queries absetzen zu können, dient somit lediglich einer ersten Überprüfung, ob die gewünschten Analysen mit dem aktuellen Modell grundsätzlich möglich sind. Für die Entwicklung komplexer Abfragen sollte ein vollwertiges Analysefrontend gewählt werden.

4.3.3 JDBC Explorer

Der lesende Zugriff auf eine relationale Datenbank wird mittels des sog. “JDBC Explorers” unterstützt (“File → New → JDBC Explorer). Es können lediglich die vorhandenen Tabellen und deren Felder (inkl. Datentypen) angezeigt werden.

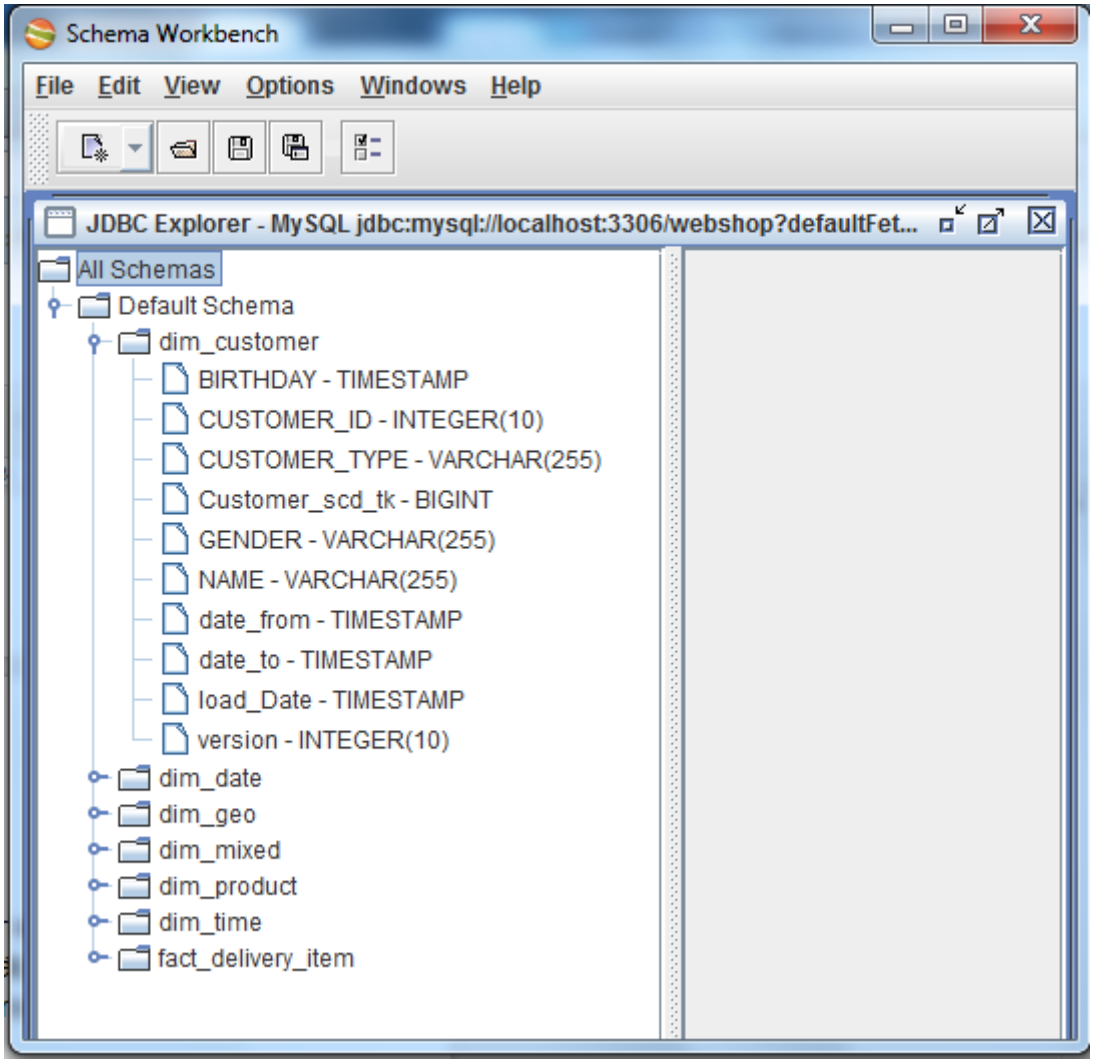


Abbildung 50 PSW JDBC-Explorer

Der JDBC-Explorer ermöglicht es, während der Modellierung des Mondrian Schemas die Strukturen des zugrunde liegenden relationalen Datenmodells nachzuschlagen.

4.4 Dokumentation

Die Schema Workbench besitzt keinerlei in die Anwendung integrierte Hilfefunktion.
Im Dateisystem im Verzeichnis der Schema Workbench (...\psw-ce-3.5.0\schema-workbench\doc) finden sich zwei PDF-Dateien ("mondrian_technical_guide.pdf" und "schema_workbench.pdf") welche eine rudimentäre und teilweise veraltete Dokumentation sowohl von Pentaho Mondrian als auch der Schema Workbench enthalten.
Eine aktuelle und umfassende Dokumentation ist auf der Mondrian-Projektseite von Pentaho verfügbar.[11] Diese enthält sowohl grundlegende Erläuterungen zu den Konzepten von OLAP und der Architektur von Pentaho Mondrian, als auch weiterführende Informationen. Dazu gehört u. a.:

- Erläuterung zum Erstellen eines Schemas (vorhandene Objekte, Struktur der Mondrian Schemata im XML-Format,
- Konfiguration des Mondrian-Servers mittels Konfigurationsdateien
- Beschreibung der API
- Optimierung (Aggregation, Cache Control)

Insgesamt erscheint die Dokumentation gut strukturiert, umfassend und vollständig, ist jedoch nur online verfügbar.

Das frei zugängliche Hilfsangebot der Pentaho Enterprise Edition enthält ebenfalls eine Anleitung zur multidimensionalen Analyse, die ein Praxisbeispiel inkl. der Modellierung eines Mondrian Schemas mit der Schema Workbench umfasst.[12]
Als eine weitere Informationsquelle dient die Community, die ein eigenes, von Pentaho betriebenes Mondrian Forum hat.[13]

[11] vgl. http://mondrian.pentaho.com/ und http://mondrian.pentaho.com/documentation/
[12] vgl.
http://infocenter.pentaho.com/help/index.jsp?topic=%2Fanalysis_guide%2Ftopic_building_rolap_cube.html
[13] vgl. http://forums.pentaho.com/forumdisplay.php?79-Mondrian-Pentaho-Analysis-Services

4.5 Umsetzung Szenario

Zur Umsetzung des Szenarios wurde ein Mondrian Schema erstellt, welches auf dem relationalen Star-Schema basiert (vgl. 1.4.3 Star-Schema). Aufgrund des multidimensionalen Ansatzes ergeben sich Unterschiede und Erweiterungen zum relationalen Modell auf die im Folgenden genauer eingegangen wird. Es wird davon ausgegangen, dass bereits eine Datenbankverbindung besteht. Die entsprechenden Zugangsdaten können in Anhang 0 nachgelesen werden. Ein Beispielschema von Pentaho befindet sich unter ...\biserver-ce\pentaho-solutions\steel-wheels\analysis\steelwheels.mondrian.xml.
Das neu erstellte Schema wurde unter dem Dateienamen “webshop_schema.xml” im Dateisystem abgelegt (siehe Anhang A0).

Im Folgenden werden lediglich die wichtigsten Eigenschaften des Modells exemplarisch anhand ausgewählter Elemente erläutert. Alles weitere kann auf dieser Grundlage am vollständigen Modell bzw. der Struktur der XML-Datei nachvollzogen werden (vgl. Anhang A3).

Als Erstes wurde in der Schema Workbench ein Cube-Element hinzugefügt und in “webshop” umbenannt. Dieses gilt Obersters Element in der Hierarchie.

4.5.1 Faktentabelle und Measures

Dem Cube werden die Elemente “Table” sowie mehrere “Measure” hinzugefügt. Das Table-Element verweist auf die Faktentabelle, die Measure-Elemente auf die gleichnamigen Kennzahlen aus dem relationalen Modell. Auf die Nutzung von zusätzlichen “Calculated Measures”, d. h. Kennzahlen, die auf Ebene von Pentaho Mondrian aus dem relationalen Modell berechnet werden, wurde verzichtet.

Für jedes Measure muss neben dem Namen und dem Verweis auf die Spalte in der Faktentabelle auch ein sog. “aggregator” festgelegt werden. Er bestimmt, wie der Feldinhalt für spätere Abfragen aggregiert wird. Zur Auswahl stehen dabei typische Aggregatsfunktionen wie SUM, COUNT, AVG etc.

Tabelle 21 Aggregatfunktionen der Measures

Measure	Aggregator
quantity	SUM
profit	SUM
returned	COUNT
shipping_cost	SUM
turnover	SUM

4.5.2 Dimensionen und Hierarchien

Dem Cube wurden sämtliche bekannte Dimensionen aus dem relationalen Schema hinzugefügt. Jede Dimension muss mindestens eine Hierarchie besitzen, um validiert zu werden. Eine Hierarchie kann ein oder mehrere Level besitzen und muss eine (Dimensions-)Tabelle referenzieren. Am Beispiel der Produkt-Dimension werden die Elemente und deren Konfiguration dargestellt:

dim_product (Dimension)

Tabelle 22 Attribute Produktdimension

Attribute	Value	Beschreibung
name	dim_product	Name der Dimension
foreignKey	PRODUCT_ID	Fremdschlüsselfeld aus Faktentabelle
type	StandardDimension	Dimensionstyp
visible	true	Sichtbarkeit

product_hierarchy (Hierarchie)

Tabelle 23 Attribute Produkthierarchie

Attribute	Value	Beschreibung
name	product_hierarchy	Name der Hierarchie
hasAll	true	Gibt an, ob die Hierarchie ein “all” Member hat
allMemberName	All Producers	Name des “all” Members
primaryKey	PRODUCT_ID	Schlüsselfeld aus der Dimensionstabelle, welches Member identifiziert und referenziert (kann erst ausgewählt werden, wenn der Hierarchie eine Tabelle zugeordnet ist!)
visible	true	Sichtbarkeit

product (Level)

Tabelle 24 Attribute Produktlevel

Attribute	Value	Beschreibung
name	product	Name des Levels
column	NAME	Name der Spalte zur Identifizierung des Levels (=Produktname)
type	String	Typ der identifizierenden Spalte
levelType	Regular	Art des Levels
hideMemberIf	Never	Bestimmt ob Level “versteckt” ist oder nicht
visible	true	Sichtbarkeit

dim_product (Table)

Tabelle 25 Attribute Tabelle Produktdimension

Attribute	Value	Beschreibung
name	dim_product	Name der Dimensionstabelle

Neben der Produkt-Hierarchie hat die Produktdimension auch noch eine Produzenten-Hierarchie mit mehreren Levels, über welche die einzelnen Produkte ausgewertet werden können. Die Level sind untereinander abhängig: Jeder Produzent kann somit einzeln oder mit Bezug zu seinen Produkten für Analysen ausgewählt werden.

Für jedes Produkt und für jeden Produzenten können so Analysen mit Bezug auf die Kennzahlen der Lieferposition in der Faktentabelle durchgeführt werden.

Degenerierte Dimension

Die Granularität der Faktentabelle wird durch eine Lieferposition repräsentiert. Um auch auf Ebene der Lieferung Auswertungen durchführen zu können, wurde eine sog. “degenerierte Dimension” erstellt. Sie enthält lediglich eine Hierarchie mit zwei Leveln, die abweichend konfiguriert sind:

dim_delivery (Dimension)

Tabelle 26 Attribute der Lieferdimension

Attribute	Value	Beschreibung
name	dim_delivery	Name der Dimension
visible	true	Sichtbarkeit

default (Hierarchie)

Tabelle 27 Attribute der default Hierarchie

Attribute	Value	Beschreibung
hasAll	true	Gibt an, ob die Hierarchie ein “all” Member hat
allMemberName	All Delivery IDs	Name des “all” Members
visible	true	Sichtbarkeit

delivery id (Level)

Tabelle 28 Attribute des Lieferlevel

Attribute	Value	Beschreibung
name	delivery id	Name des Levels
column	delivery_id	Name der Spalte zur Identifizierung des Levels (=Produktname), muss manuell eingegeben werden, bezieht sich auf Spalte aus Faktentabelle
type	String	Typ der identifizierenden Spalte
uniqueMembers	true	Gibt an, ob Member über alle Parents einzigartig sind
levelType	Regular	Art des Levels
hideMemberIf	Never	Bestimmt ob Level “versteckt” ist oder nicht
visible	true	Sichtbarkeit

delivery item id (Level)

Tabelle 29 Attribute Lieferpositionlevel

Attribute	Value	Beschreibung
name	delivery item id	Name des Levels
column	delivery_item_id	Name der Spalte zur Identifizierung des Levels (=Produktname), muss manuell eingegeben werden, bezieht sich auf Spalte aus Faktentabelle
type	String	Typ der identifizierenden Spalte
uniqueMembers	true	Gibt an, ob Member über alle Parents einzigartig sind
levelType	Regular	Art des Levels
hideMemberIf	Never	Bestimmt ob Level “versteckt” ist oder nicht
visible	true	Sichtbarkeit

Die degenerierte Dimension “dim_delivery” enthält keine Tabelle und bezieht sich daher auf die Faktentabelle. Durch die beiden Level der Hierarchie kann auf Ebene der Lieferung aggregiert werden. Weiterhin können auch alle Positionen einer Lieferung ausgewertet werden. Aufgrund der fehlenden Tabelle wird die Dimension nicht validiert. Das Modell wird von der Schema Workbench als fehlerhaft angesehen, obwohl es funktionsfähig ist. Der Validierungsalgorithmus kennt den Sonderfall der degenerierten Dimension nicht, Pentaho Mondrian verarbeitet das Schema jedoch problemlos und mit der gewünschten Funktionalität.[14]

[14] vgl. http://mondrian.pentaho.com/documentation/schema.php#Degenerate_dimensions

4.5.3 Publishing auf Server

Um das Schema als Datenquelle auf dem BI-Server verwenden zu können, muss es zunächst veröffentlicht werden. Dafür muss das Publisher-Passwort auf dem Server gesetzt sein und eine Datenbankverbindung in der Administrator Console eingetragen sein.[15] Danach kann das Schema mittels “File → New → Publish” auf den Server kopiert werden:

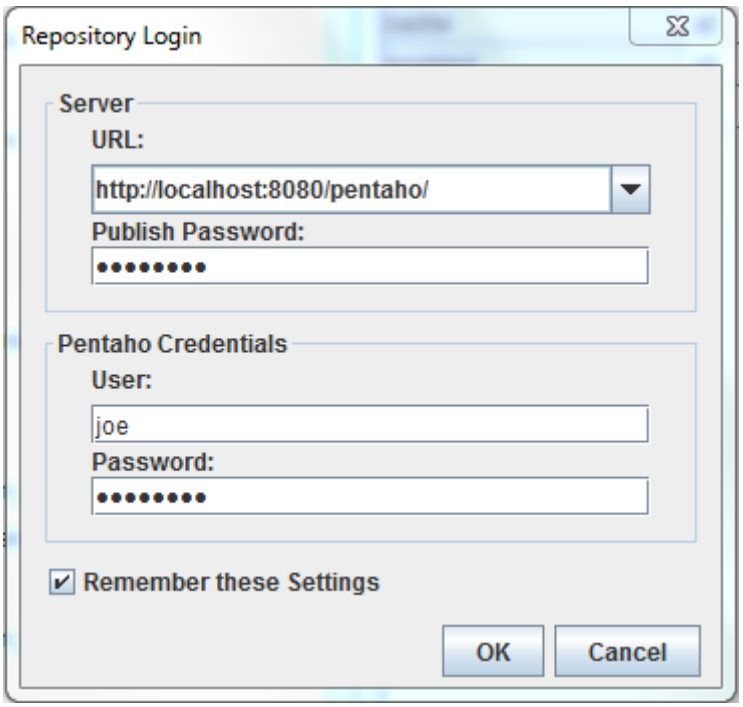


Abbildung 51 PSW Repository Login

Das aktuelle Publish Password sowie URL und die Pentaho Credentials richten sich nach der jeweiligen Konfiguration des BI-Servers in der virtuellen Maschine (siehe Anhang A0). Nach dem Veröffentlichen steht das Schema auf dem Server für die Nutzung durch die dort integrierten OALP-Frontends zur Verfügung.

[15] vgl. http://wiki.pentaho.com/display/ServerDoc1x/Configuring+Publish

5 Reporting

5.1 Übersicht

Der Report Designer ist ein Stand Alone Tool. Die aktuellste Version und verwendete stabile Version ist 3.9.1. Bei der Standardvariante von Pentaho muss der Report Designer extra heruntergeladen werden, entpackt und einfach gestartet werden. Zum Report Designer gibt es einen User Guide in englischer Sprache, wo unter anderem Tutorials enthalten sind. Der Report Designer von Pentaho ist schnell und intuitiv zu bedienen. Das Standard-Vorgehen zur Erstellung eines Reports ist folgendes: Als erstes ist eine Connection zu der Datenbank anzulegen. Auf die Datenbank wird dann eine Query erstellt. Die Query ist wiederum Grundlage des Berichts. Es ist möglich, auf einem bestimmten Repository zu arbeiten. Das Dateiformat für die Reports ist .prpt und basiert auf XML.

5.2 Oberfläche und Bedienbarkeit

Nach Aufruf der “report_designer.bat”[16] offeriert der Welcome Screen zunächst eine Auswahl an Beispielreports, sowie die Möglichkeit einen neuen Report zu erstellen, oder den Report Wizard in Anspruch zu nehmen.

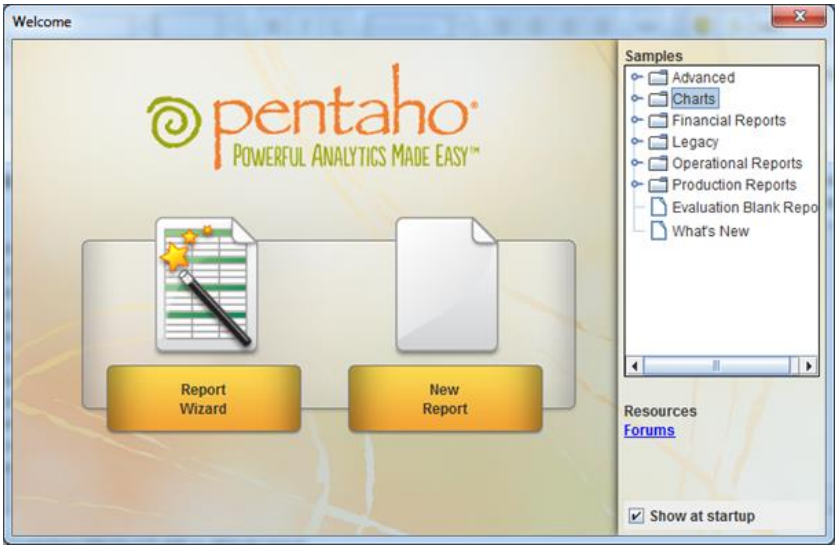


Abbildung 52 PRD Startbildschirm

[16] Server Dateipfad: C:\PDWH\pentaho\prd-ce-3.9.1\report-designer

Der Report Wizard unterstützt den Anwender bei der Erstellung von Berichten durch vorgefertigte Templates. Die zur Verfügung gestellten Beispielreports sind umfassend und schnell und einfach zu erstellen.

Die Oberfläche des Reportdesigners ist strukturiert aufgebaut. In der Mitte befindet sich der Modellierungsbereich, Links daneben die Modellierungsleiste mit den verfügbaren Modellierungsobjekten. Unter dem Modellierungsbereich wird ein Fehlerlog angezeigt, womit bereits frühzeitig Warnungen und Fehler angezeigt werden.

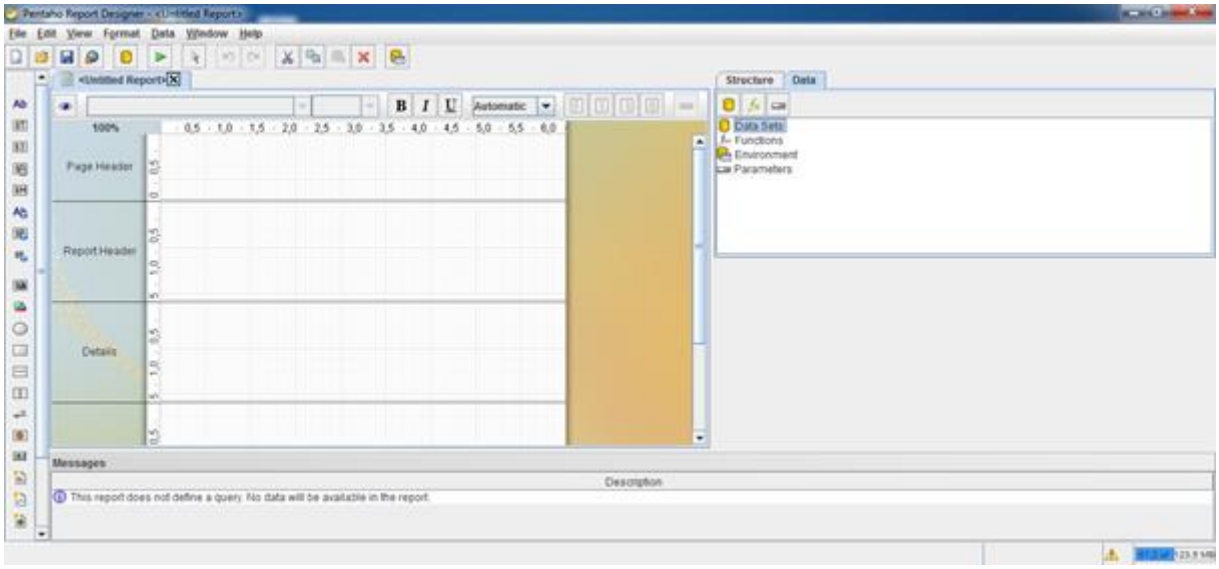


Abbildung 53 PRD Übersicht Arbeitsbereich

Auf der rechten Seite im Report Designer gibt es die Reiter Structure und Data.
Der Reiter Structure zeigt die Steuerelemente und ihre Anordnung im jeweiligen Bereich des Berichtes. Der Reiter Data enthält das Data Set mit der jeweiligen Connection und den Feldern der entsprechenden Query. Um eine Query als Grundlage eines Reports verwenden zu können, ist diese vorher auszuwählen.

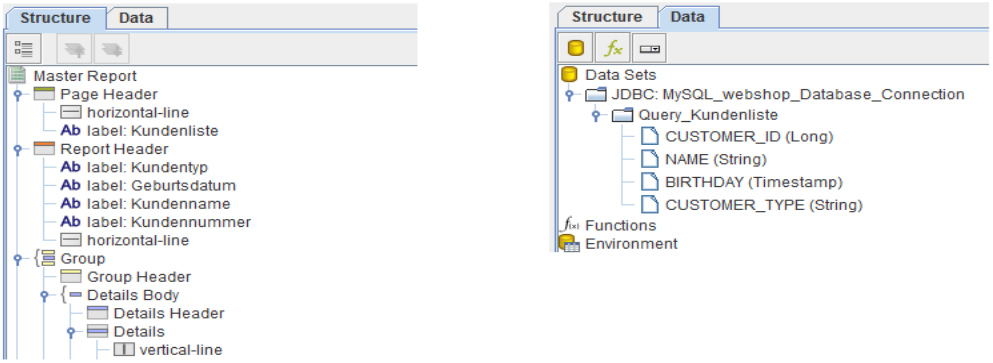


Abbildung 54 PRD Structure/Data Reiter

Grundsätzlich sind Reports aufgebaut aus Zeilenkopf, Reportkopf, Detailbereich, Reportfußbereich und einen Seitenfußbereich. Der Benutzer wird bei der Ausrichtung der Steuerelemente mit Inch-Angaben(können auf cm-Angaben geändert werden) und karierten Hintergrund unterstützt. Die Ausrichtung der Steuerelemente im Bericht ist wesentlich.

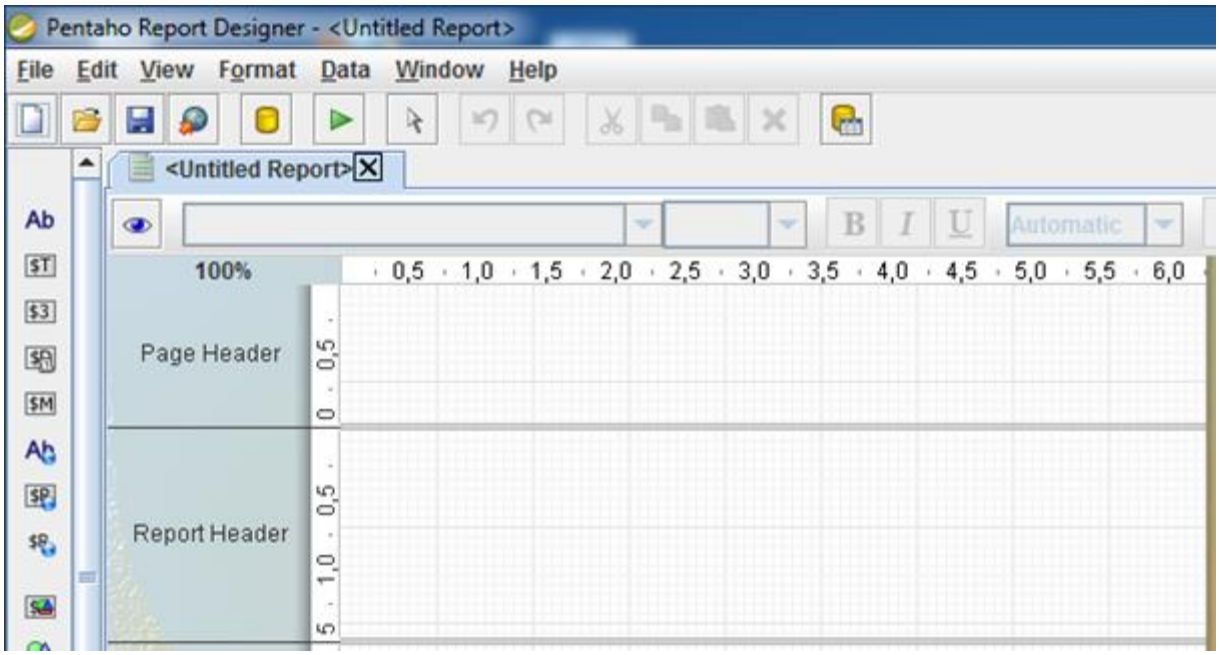


Abbildung 55 PRD Modellierungsbereich

Über die Parameter können SQL-Statements parametrisiert werden. Dabei können die Parameter vom Benutzer beim Erstellen eines Diagramms vorher abgefragt werden, um so z. B. einen bestimmten State (US-Bundesstaat) oder eine Jahreszahl auszuwählen.

5.3 Funktionalität

Zu den Standard-Steuerelementen, die am häufigsten verwendet werden, zählen Textfelder und Bezeichnungsfelder und für Formatierungen sind es das Band und die Linien-Elemente. Mit den Ressource-Elementen sind mehrsprachige Reports möglich, die Labels werden dann entsprechend in der ausgewählten Sprache ausgegeben.

Tabelle 30 PRD Übersicht Steuerelemente

Steuerelement	Beschreibung
	Bezeichnungsfeld (Überschriften können hinzugefügt werden)
	Textfeld (Feldinhalte aus der Datenbank können abgebildet werden.)
	Zahlenfeld
	Datumsfeld
	Meldung
	resource-label
	resource-field
	resource-message
	Imagefield (Images können den Berichten hinzugefügt werden.)
	Imagefield
	Kreis (Kreise, Rechtecke und Linien dienen dazu Bereich oder Tabellen zu erzeugen oder einzukreisen.)
	Rechteck
	Linie

	Tabelle
	Survey-scale
	Charts
	Barcode
	Balkendiagramm
	Liniendiagramm
	Kreisdiagramm
	Band (Mit dem Band können einheitliche Hintergründe definiert werden.)
	Es können in Berichten Unter(Sub-)Berichte erstellt werden.
	Erstellung eines Inhaltsverzeichnisses
	Erstellung eines Indexes ist möglich.

Ein Export ist als .pdf, .html, .xls, .xlsx, .rtf, .txt und .csv Datei möglich. Des Weiteren lässt sich der Report auch direkt ausdrucken.

Als Datengrundlage besteht eine Auswahl zwischen Advanced Database-Sources: JDBC(Custom), Java Script, Named Java Method Invocation, Hibernate und External.

Beim Erstellen von Reports wird der Benutzer über mehrere Auswahlmöglichkeiten zum Erstellen von Berichten unterstützt. Der Benutzer wählt eine mögliche Variante von Look and Feel, fügt Einstellungen zur Data Source hinzu und passt das Layout und das Format an.

SQL-Kenntnisse sollten vorhanden sein, da die verwendeten Variablen für die Diagramme grundlegend aus berechneten Funktionen (min, max, sum usw.) berechnet werden und die Grundlage bilden.

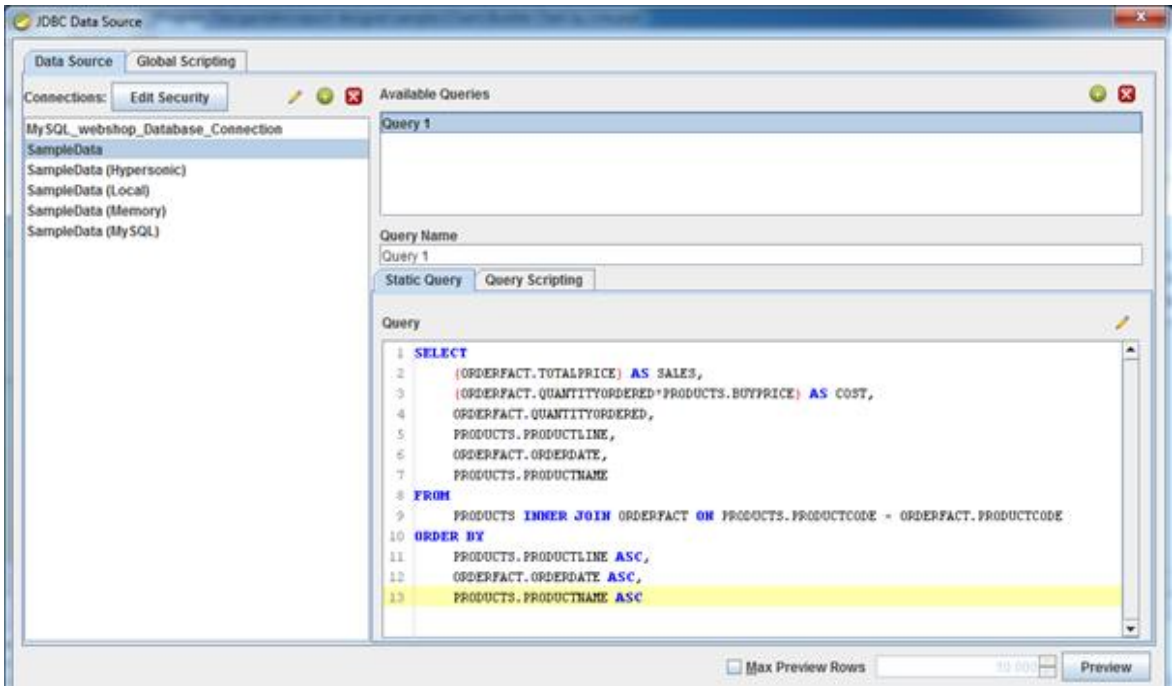


Abbildung 56 PRD Connectionmanager

Neben SQL-Statements können auch MDX-Statements verwendet werden.
Dazu ist eine Verbindung zur Datenbank herzustellen und die Schema-Datei mit einzubinden. (Pfad auf dem Server: C:\PDWH\pentaho\olap\webshop_schema.xml)

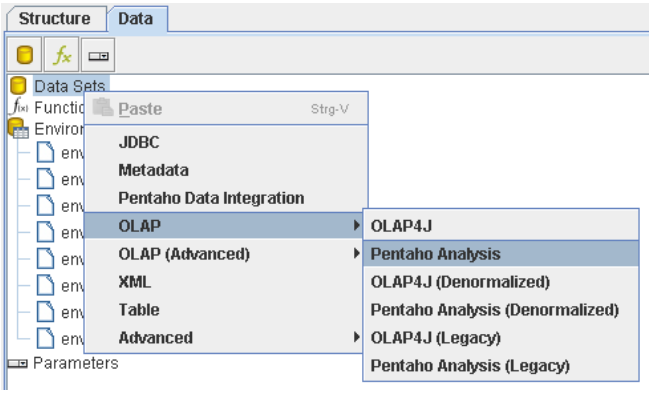


Abbildung 57 PRD Verbindungstypen

Auf dem Cube Webshop können nun MDX-Abfragen ausgeführt werden.

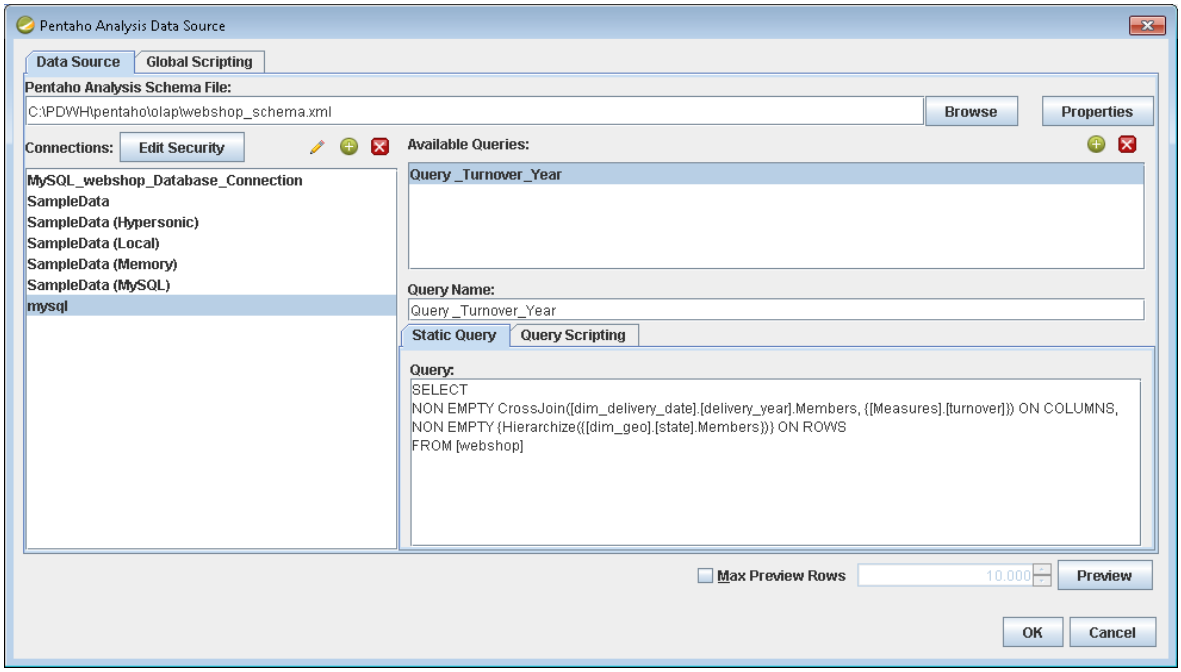


Abbildung 58 PRD Query Management

Das Chart-Steuerelement enthält eine große Anzahl an verschiedenen Diagrammtypen (Balken-, Linien-, Flächen-, Kreisdiagrammen uvm.).

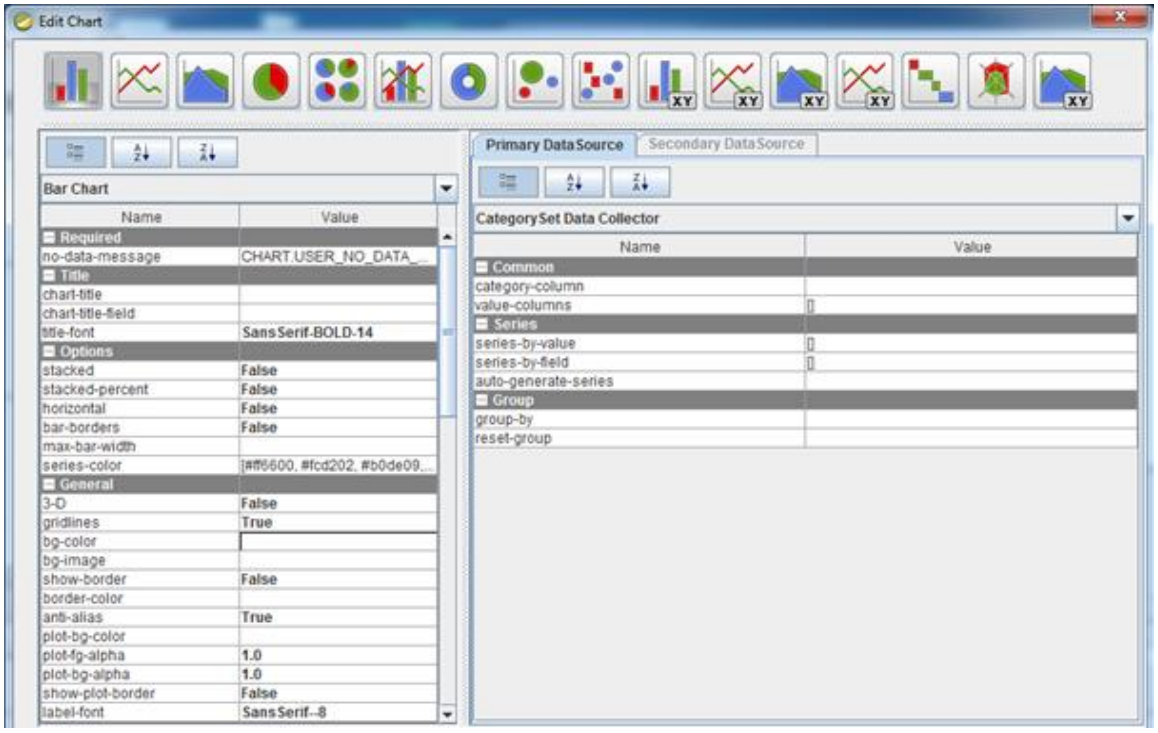


Abbildung 59 PRD Übersicht Edit Chart

5.4 Dokumentation

Eine Beschreibung der grundlegenden Möglichkeiten des Report Designers und eine Installationsanleitung befindet sich in der Pentaho Wiki.[17] Dieses Wiki enthält eine Installationsanleitung und eine Beschreibung der Arbeitsumgebung. Es wird beschrieben wie Reports mit dem Report Wizard erstellt werden können. Das Anlegen von Data Sources wird beschrieben und es gibt eine Beschreibung zu den grafischen Elementen. Zudem gibt es die Möglichkeit mehrsprachige Reports zu generieren und es können Sub-Reports erstellt werden. Zusätzlich gibt es Hinweise zum Veröffentlichen einzelner Reports mit der Verwendung von Parametern und die Anwendung von integrierten Funktionen, wie z. B. mathematische, logische, Datums- und Text-, sowie Informations-Funktionen. Bei den Informationsfunktionen können Null-Werte abgefragt werden. Ein digitales Benutzerhandbuch ist auch im PDF-Format vorhanden, welches in einem bestimmten Pfad abgelegt und darauf aus dem Report Designer verknüpft werden kann.[18] Die Dokumentation liegt in englischer Sprache vor.

5.5 Umsetzung Szenario

Anhand von 3 Szenarien werden die grundlegenden Funktionen des Reportdesigners getestet.
Zunächst soll im Szenario 1 eine einfache gefilterte SQL Query dargestellt und formatiert werden. Im Szenario 2 soll beispielhaft dargestellt werden, wie eine Visualisierung mittels Diagrammen erfolgen kann. Szenario 3 beschreibt die Filterung mittels Parameter und aufrufen eines Sub-Reports. Die in den Szenarien generierten MDX-Queries liegen im Anhang A6 vor.

[17] Siehe: http://wiki.pentaho.com/display/Reporting/Report+Designer#
[18] Server Dateipfad: \pentaho\report-designer\docs\report_designer_user_quide.pdf

5.5.1 Szenario 1 Erstellung einer Kundenliste

Die Kundenliste soll alle Geschäftskunden mit dem Anfangsbuchstaben M, die später als 1980 geboren wurden und einem Kundennummernbereich zwischen 10000001500 und 10000002000 enthalten. Dazu wird für die Data Source eine neue Query angelegt.

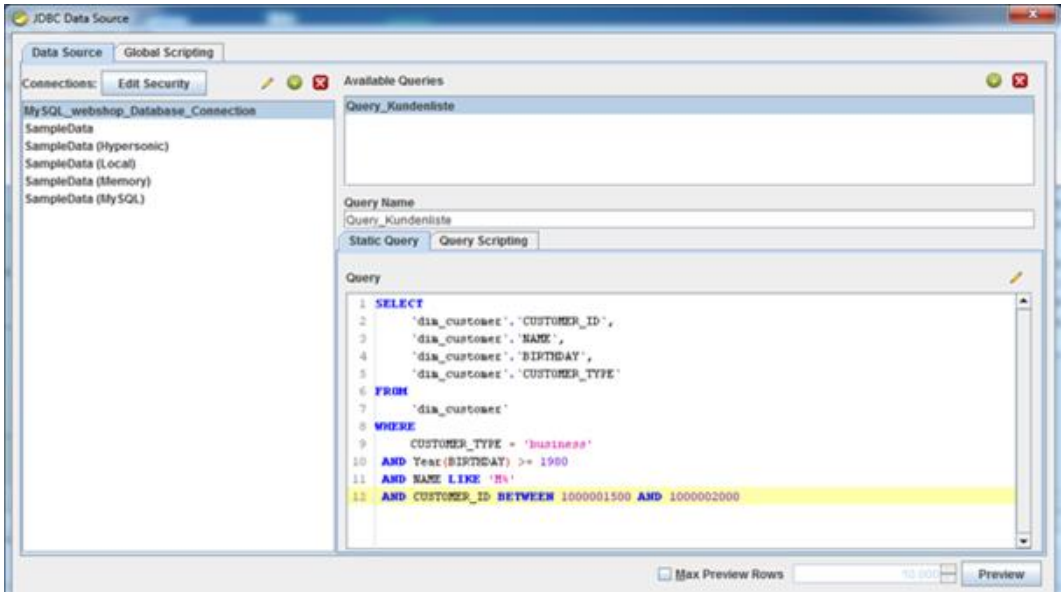


Abbildung 60 PRD Szenario 1 Query

Auf der linken Seite wird der Bericht mit den verwendeten Steuerelementen dargestellt und kann nach Fertigstellung schnell in einer Druckansicht (Seitenvorschau) mit den Ergebnissen der Query am Bildschirm angezeigt werden.

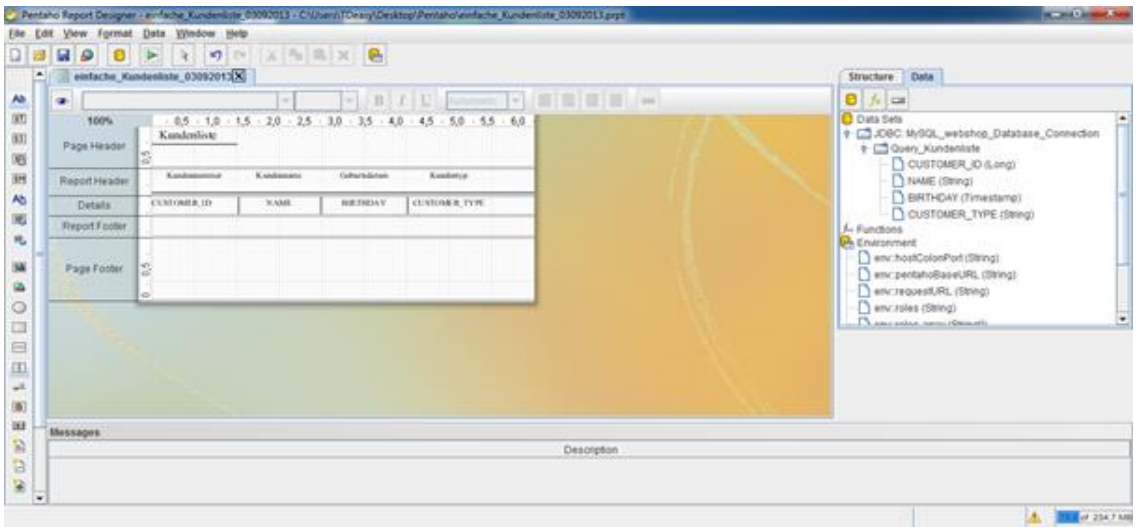


Abbildung 61 PRD Szenario 1 Reportentwurf

Die Kundenliste enthält die Felder Kundennummer, den Kundennamen, das Geburtsdatum und den Kundentyp. Über die Where-Klausel der SQL-Query kann das Data Set eingeschränkt werden, um kundenindividuelle Berichte zu erstellen.

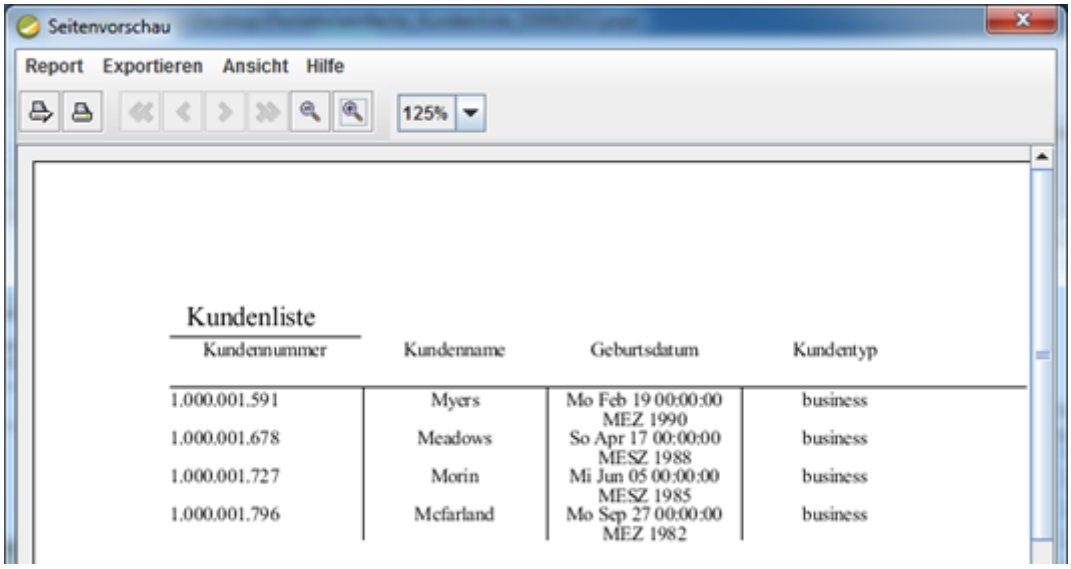

Seitenvorschau
Report Exportieren Ansicht Hilfe
125%
Kundenliste
Kundennummer Kundenname Geburtsdatum Kundentyp
1.000.001.591 Myers Mo Feb 19 00:00:00 MEZ 1990 business
1.000.001.678 Meadows So Apr 17 00:00:00 MESZ 1988 business
1.000.001.727 Morin Mi Jun 05 00:00:00 MESZ 1985 business
1.000.001.796 Mcfarland Mo Sep 27 00:00:00 MEZ 1982 business

Abbildung 62 PRD Szenario 1 Ausgabe unformatiert

Anschließend können noch Formatierungen, wie hier etwa eine grau Schattierung und Formatänderung für die Kundennummer und die Anzeige des Geburtsdatums vorgenommen werden.

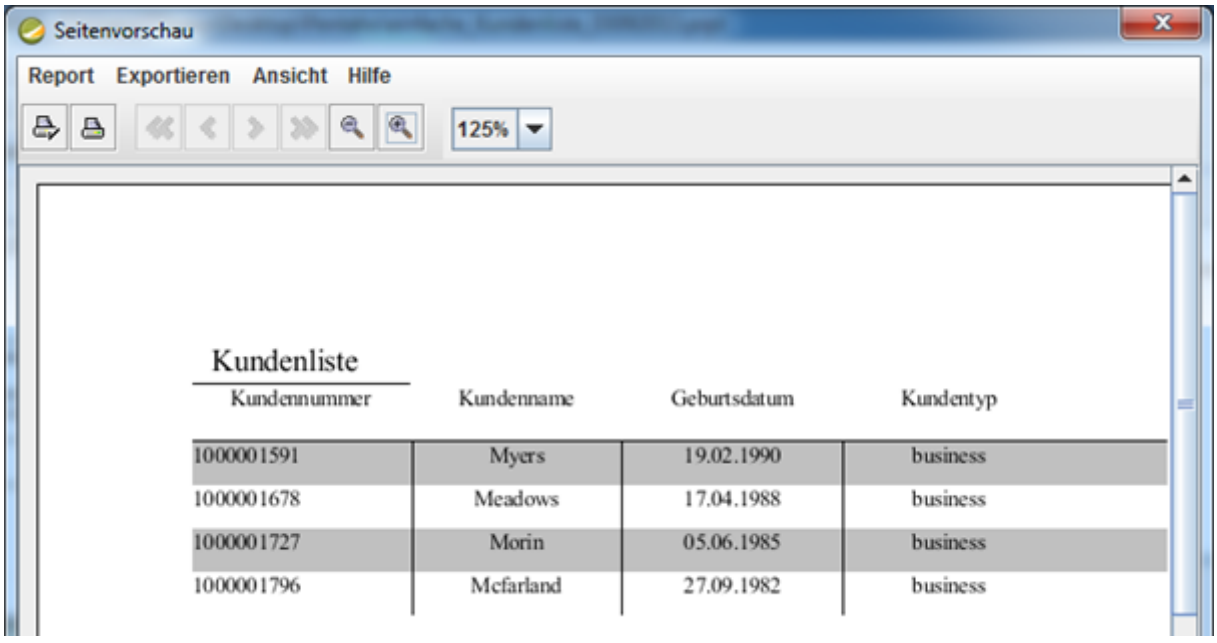


Abbildung 63 PRD Szenario 1 Ausgabe formatiert

5.5.2 Szenario 2 Darstellungen mittels Diagramme

Als Beispiele für die Diagramme wurden ein Balken- und ein Kreisdiagramm erstellt. Im Kreisdiagramm wird der Gewinn summiert auf die fünf Produktgruppen dargestellt. Im Balkendiagramm wird der Umsatz je Produktgruppe und durchgeführter Versand und Transport über das entsprechende Transport- und Logistikunternehmen dargestellt.

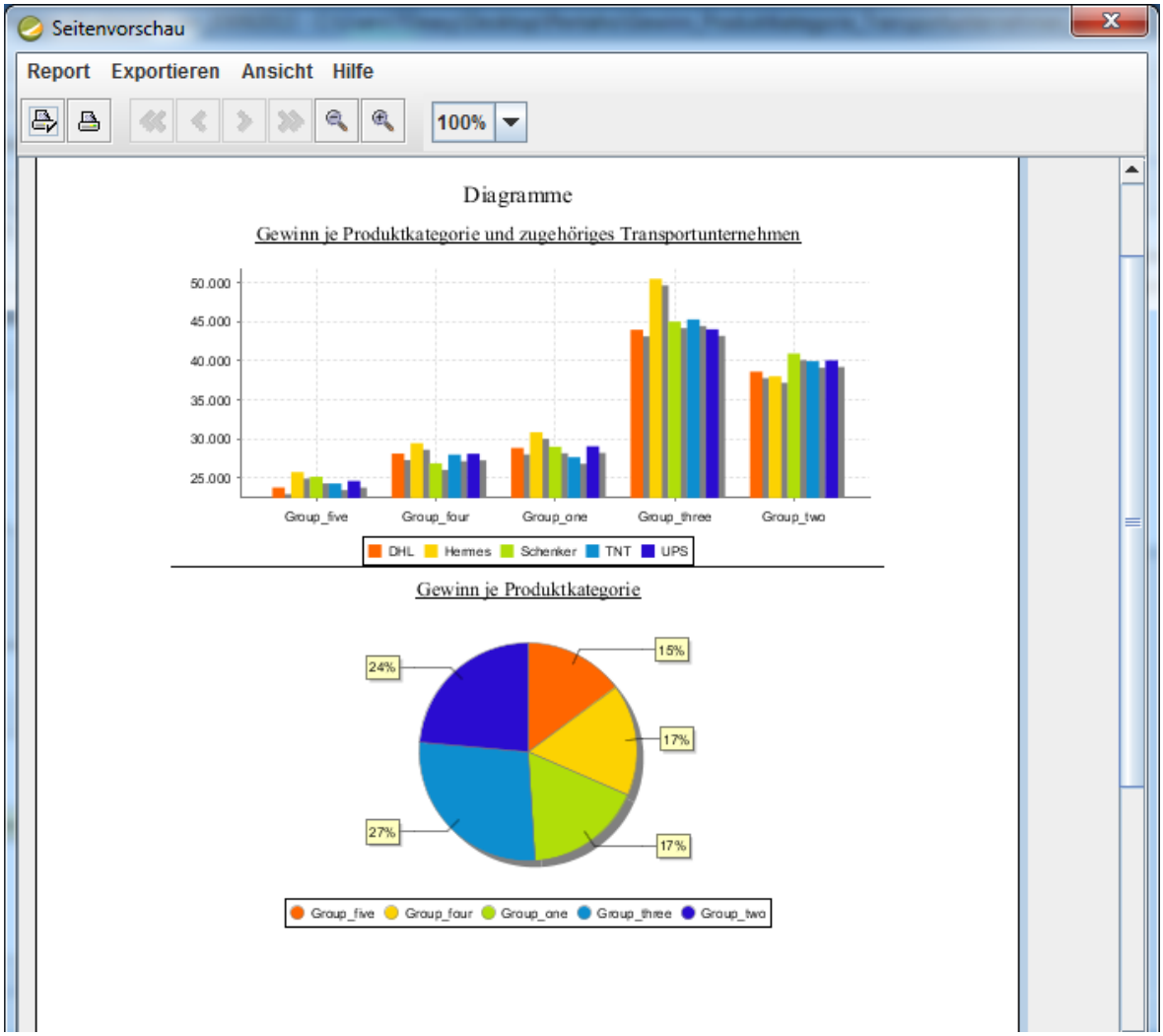


Abbildung 64 PRD Ausgabe Szenario 2 Diagramme

Die Prozentwerte des Kreisdiagramms werden automatisch berechnet, indem der komplette Kreis den Gesamtgewinn darstellt und die Werte je Produktkategorie ins Verhältnis gesetzt werden. Die Grundlage beider Charts ist folgendes SQL-Statement.

Query Name

Query_Gewinn_Produktgruppe_Versand

Static Query | Query Scripting

Query

```
1
2
3  SELECT
4       Sum(`fact_delivery_item`.`PROFIT`) AS Gewinn,
5       `dim_mixed`.`SHIPPING_COMPANY`,
6       `dim_mixed`.`PRODUCT_GROUP`
7  FROM
8       `dim_mixed` INNER JOIN `fact_delivery_item` ON `dim_mixed`.`MIXED_ID` = `fact_delivery_item`.`MIXED_ID`
9  GROUP BY
10      `dim_mixed`.`PRODUCT_GROUP`,
11      `dim_mixed`.`SHIPPING_COMPANY`
```

Abbildung 65 PRD Query Szenario 2

5.5.3 Szenario 3 Filtern mittels Parameter

Mit Parameter in der SQL-Abfrage kann die Abfrage flexibel durch die Eingabe des Benutzers gesteuert werden, um so das Data Set zu verändern durch die Bedingung in der Where-Klausel. Der Parameter wird durch ${parameter} in der Where-Klausel hinzugefügt.

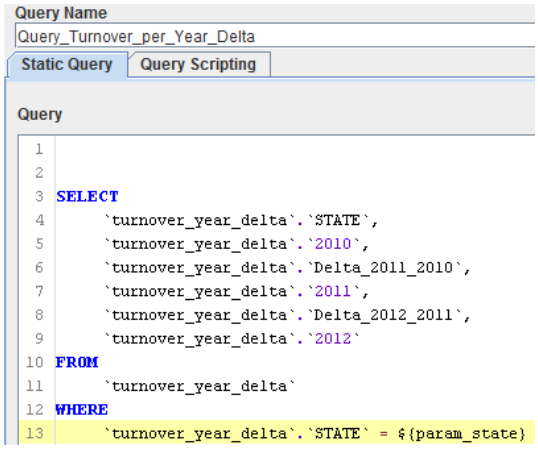


Abbildung 66 PRD Query Szenario 3 Filter

Als Sub-Report wurde hier ein Report mit einem Balkendiagramm verwendet.

Abbildung 67 PRD Reportentwurf Szenario 3

Der Parameter wird zwischen dem Sub-Report und dem eigentlichen Report zugeordnet, d. h. der Wert des Parameters wird dem Sub-Report übermittelt und das Data Set des Sub-Reports basiert dann nur noch auf dem Wert des Parameters.

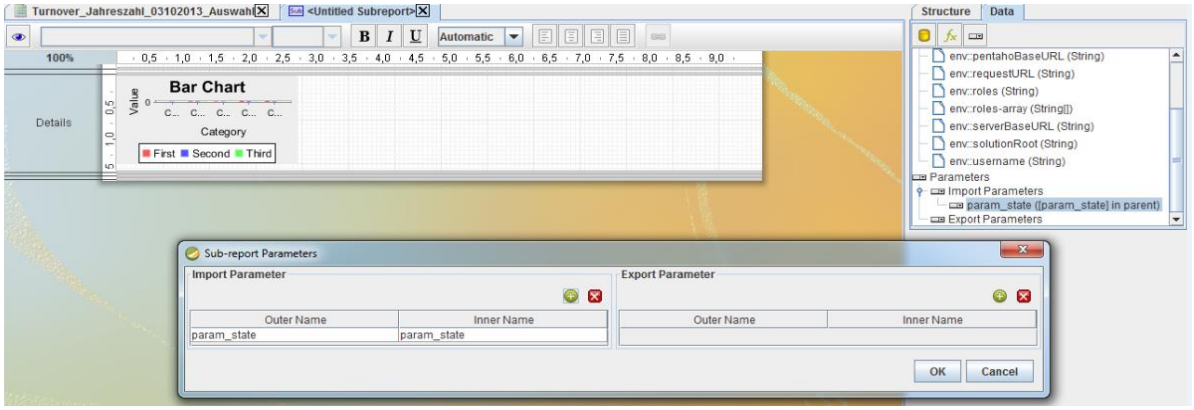


Abbildung 68 PRD Szenario 3 Verknüpfung des Import-Parameters

Beim Ausführen des Reports mit den entsprechenden Daten, wird dem Benutzer die Möglichkeit gegeben ein State auszuwählen. Hier im Beispiel wird “California” ausgewählt, aus allen möglichen US-Bundesstaaten.

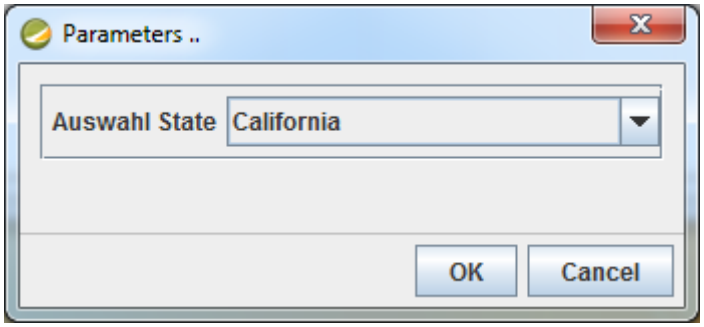


Abbildung 69 PRD Szenario 3 Parameterauswahl

Im Balkendiagramm werden die Umsätze der einzelnen Jahre (2010, 2011 und 2012) dargestellt. Zusätzlich wurde die Differenz zwischen den Jahren berechnet.

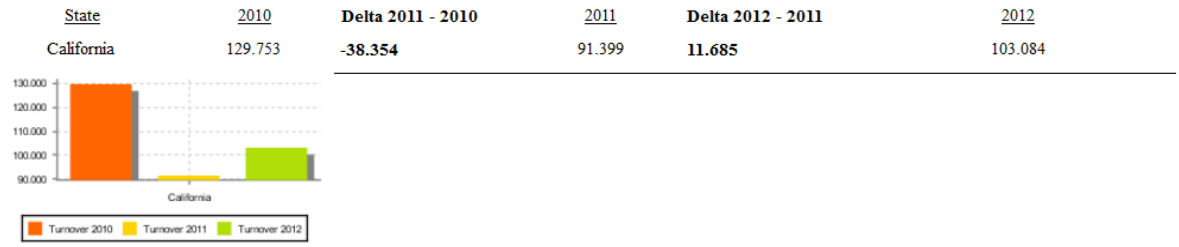


Abbildung 70 PRD Szenario 3 Auswahlergebnis

Im SQL-Script wird eine Delta-Berechnung zwischen den Jahreszahlen durchgeführt.

MySQL kennt keinen Pivot-Befehl, um Spalten und Zeilen zu vertauschen und entsprechend zu summieren. Durch die Verwendung von Views und einer Anwendung von IF-Bedingungen im SQL-Script (siehe Turnover_Year_Pivot.sql) wurden die Ergebnisse erstellt. Einmal wurden die Reports mit einer MDX-Abfrage und einmal nur mit SQL erstellt. Am Beispiel wird dies für den nächsten Report beschrieben (siehe Anhang: A2 SQL Grundlage für Reports).

Der Umsatz wird je Jahreszahl (2010, 2011, 2012) je State berechnet.

Als Ergebnis wird eine Liste mit den Umsätzen je Zeile erstellt. Ziel ist es Spalten und Zeilen mit SQL zu vertauschen, dies wäre einfach zu lösen, indem ein Pivot-Befehl verwendet werden könnte.

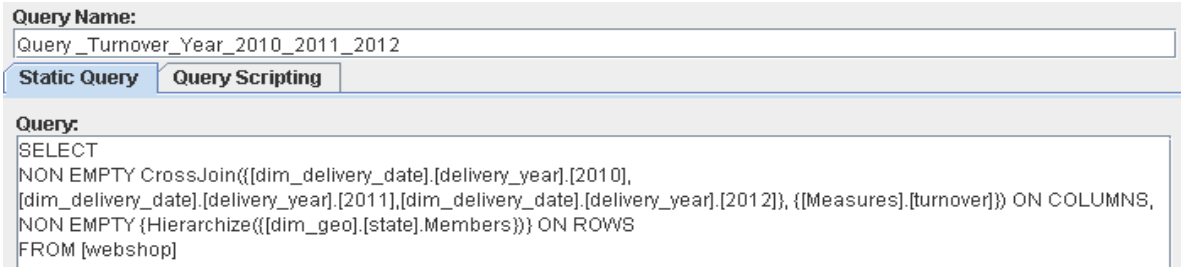


Abbildung 71 PRD Szenario 3 MDX Query

Die MDX-Abfrage ist schnell zu erstellen und mit MDX können komplizierte Abfragen einfacher erstellt werden.

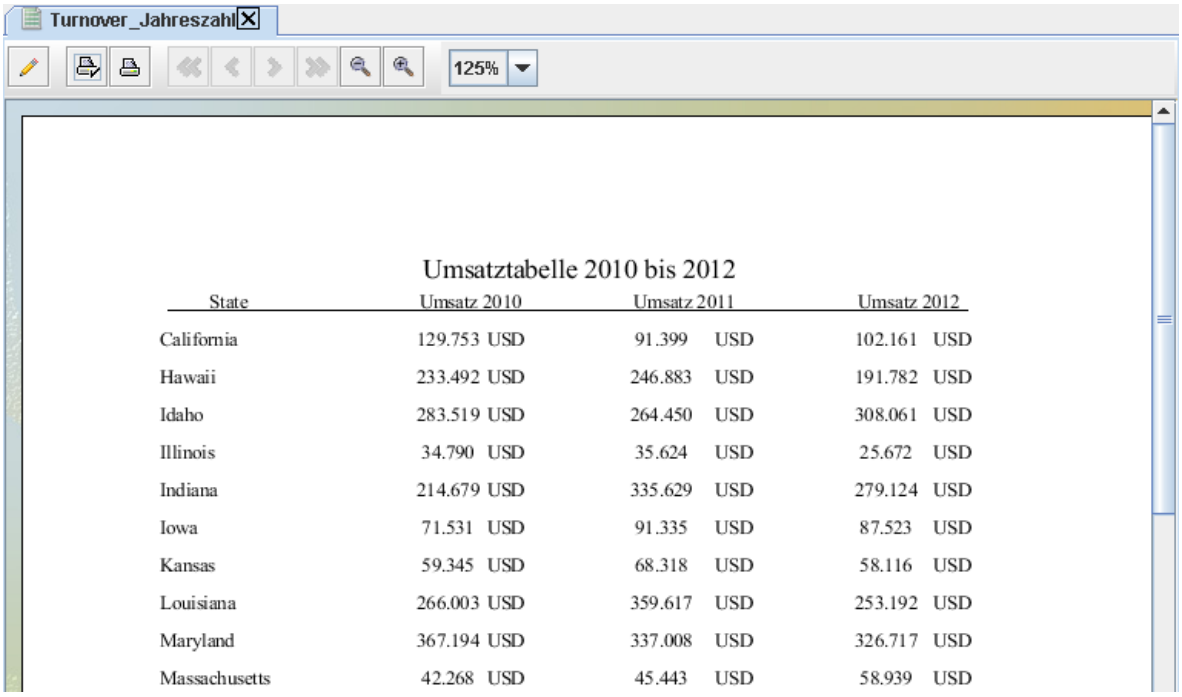

Turnover_Jahreszahl
125%
Umsatztabelle 2010 bis 2012
State
Umsatz 2010
Umsatz 2011
Umsatz 2012
California 129.753 USD 91.399 USD 102.161 USD
Hawaii 233.492 USD 246.883 USD 191.782 USD
Idaho 283.519 USD 264.450 USD 308.061 USD
Illinois 34.790 USD 35.624 USD 25.672 USD
Indiana 214.679 USD 335.629 USD 279.124 USD
Iowa 71.531 USD 91.335 USD 87.523 USD
Kansas 59.345 USD 68.318 USD 58.116 USD
Louisiana 266.003 USD 359.617 USD 253.192 USD
Maryland 367.194 USD 337.008 USD 326.717 USD
Massachusetts 42.268 USD 45.443 USD 58.939 USD

Abbildung 72 PRD Szenario 3 MDX Ausgabe

Als Ergebnis wird eine Liste mit den Umsätzen je Zeile erstellt.
Es werden zusätzlich die Abfragen als Views gespeichert, um später auf diese referenzieren zu können.

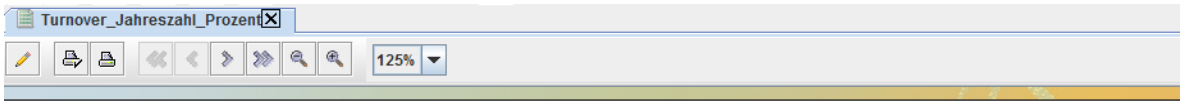
Um die Prozentwerte berechnen zu können, sollten die Gesamtumsätze berechnet werden, um die einzelnen Umsätze der US-Bundesstaaten zu den Gesamtumsätzen ins Verhältnis setzen zu können.

MySQL unterstützt nicht die Verwendung eines Pivot-Befehls, sondern hier wird eine IF-Bedingung mit einer Sum-Funktion kombiniert.

Für die Delta-Berechnung und die Prozentwerte wurde eine weitere View angelegt.

Die gespeicherten Views sind zum einen die Umsätze je State(turnover_year) und zusätzlich die Berechnung zwischen den Jahreswerten (turnover_year_delta).

Als Ergebnis wird ein Report mit einer Liste erstellt, in der die Prozentwerte aufgelistet werden. Es wurde nur noch die Währung und das Prozentzeichen als Label hinzugefügt.



		Gesamtumsatz 2010: 6.730.317 USD		Gesamtumsatz 2011: 6.916.811 USD		Gesamtumsatz 2012: 6.844.933 USD
California	1,93 %	129.753 USD	1,32 %	91.399 USD	1,51 %	103.084 USD
Hawaii	3,52 %	237.222 USD	3,52 %	243.153 USD	2,83 %	193.635 USD
Idaho	4,21 %	283.519 USD	3,82 %	264.450 USD	4,5 %	308.061 USD
Illinois	0,52 %	34.790 USD	0,52 %	35.624 USD	0,38 %	25.672 USD
Indiana	3,28 %	220.558 USD	4,79 %	331.124 USD	4,06 %	277.750 USD
Iowa	1,06 %	71.531 USD	1,32 %	91.335 USD	1,28 %	87.523 USD
Kansas	0,88 %	59.345 USD	0,99 %	68.318 USD	0,85 %	58.116 USD
Louisiana	3,95 %	266.003 USD	5,2 %	359.617 USD	3,7 %	253.192 USD
Maryland	5,49 %	369.616 USD	4,84 %	334.586 USD	4,77 %	326.717 USD
Massachusetts	0,63 %	42.268 USD	0,66 %	45.443 USD	0,86 %	58.939 USD
Minnesota	2,74 %	184.080 USD	3,01 %	207.997 USD	3,27 %	223.977 USD
Mississippi	3,66 %	246.559 USD	3,46 %	239.307 USD	3,38 %	231.085 USD

Abbildung 73 PRD Szenario 3 SQL Ausgabe

Die Lösung dieses Reports basiert auf SQL-Abfragen, hier könnten auch MDX-Abfragen verwendet werden. Die Prozentwerte entsprechen dem Anteil am Gesamtumsatz des entsprechenden Jahresumsatzes.

6 Visualisierung und Dashboards

6.1 Übersicht

In der vorliegenden Community Edition hat Pentaho keine aktiv unterstützten Dashboard Funktionen. Bis 2008 wurde zur Modellierung von Dashboards JPivot genutzt. Es folgte der Wechsel in der Enterprise Edition auf ClearView von LucidEra. 2009 wurde LucidEra von Pentaho übernommen und vollständig integriert.[19] Ein unterstützter Ersatz für die Community Edition direkt von Pentaho gibt es nicht. Mittels des Marketplace können entsprechenden Funktionen hinzugefügt werden.

Der Marketplace bietet zwei Plugins die Dashboard Funktionen implementieren. Zum einen den Community Dashboard Editor von Webdetails und zum anderen Saiku von Analytical Labs. Wie bereits im Kapitel 2.2 beschrieben wurde Webdetails im April 2013 von Pentaho aufgekauft worden. Da zum derzeitigen Zeitpunkt nicht klar ist, ob die Plugins von Webdetails noch in der kommenden Community Edition verfügbar sind, wird in dieser Ausarbeitung nur Saiku betrachtet.

Saiku wurde von 2008 an als Pentaho Analysis Tool entwickelt und 2010 als Saiku veröffentlicht. Saiku ist zum einen als Stand Alone Tool vorhanden, zum anderen auch als Plugin für u. a. Pentaho. Es verwendet die OLAP4J 1.1.0 API und ist durch XML/A kompatibel zu Mondrian 4+, MS SQL SSAS, PALO und SAP BW.[20]

Saiku liegt in der Plugin-Version 2.5 vor. Saiku verwendet als eigenes Dateiformat .saiku, welches auf XML basiert. Neben der Speicherung der Ergebnisse im .saiku Format in der Ordnerstruktur auf dem Server, können auch Exporte durchgeführt werden. Diese können in den Formaten .xls, .csv und .pdf erfolgen.

Auf http://dev.analytical-labs.com/ ist zudem stets die aktuellste Version als Livetest verfügbar.

[19] Siehe: http://www.informationweek.com/software/soa-webservices/pentaho-buying-saas-from-lucidera/220301067
[20] Siehe: http://www.olap4j.org/

6.2 Oberfläche und Bedienbarkeit

Die Oberfläche ist klar strukturiert und leicht zu bedienen. Auf der linken Seite erfolgt die Auswahl der Datenquelle und Auflistung der Dimensionen und Kennzahlen. Diese werden mittels Drag&Drop in den Modellierungsbereich (Spalten/Zeilen/Filter) gezogen. Direkt unter dem Modellierungsbereich erfolgt die Ausgabe der Generierung. Rechts von der Ausgabe kann der Darstellungsmodus ausgewählt werden. Dabei kann zwischen Tabellarischen und Grafischen Modus gewechselt werden. Im grafischen Modus können verschiedene Diagrammtypen ausgewählt werden. Diese werden automatisch mit den Daten der Tabelle gefüllt und ausgegeben. Im tabellarischen Modus können in der Tabelle zusätzlich Sparklines automatisch eingefügt werden. Zur Überprüfung der Daten können zudem einfache statistische Kennzahlen ausgegeben werden.

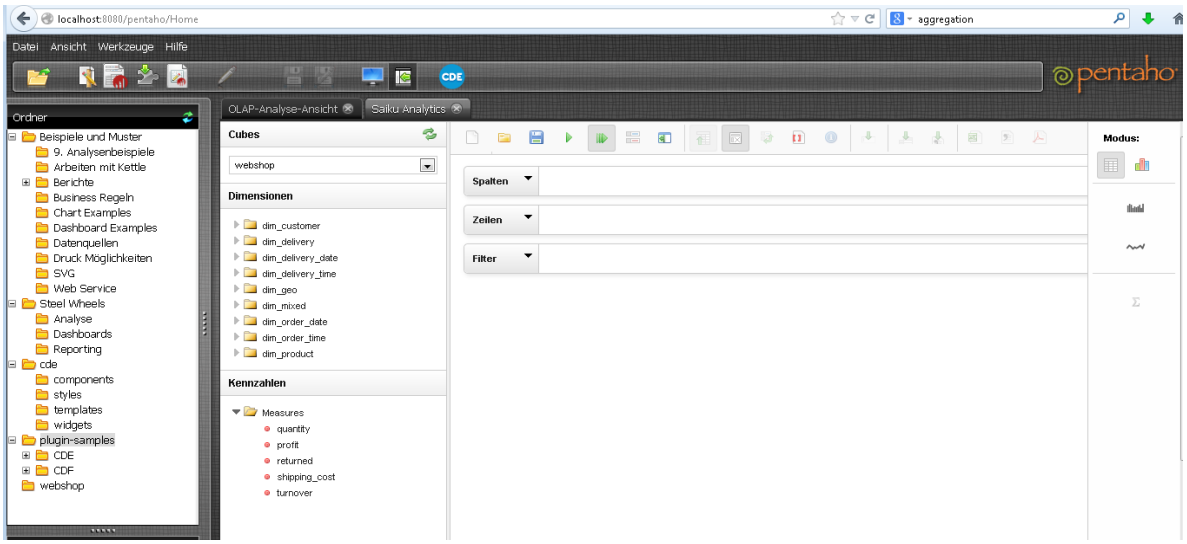


Abbildung 74 Saiku Startbildschirm

Die Oberfläche basiert vollständig auf HTML und Javascript und ist somit leicht individualisierbar. Des Weiteren lassen sich alle Bereiche ausblenden, sodass nur der Ausgabebereich sichtbar ist.

6.3 Funktionalität

Die Hauptfunktionen zur Erstellung von Dashboards mittels Saiku lassen sich in die drei Gruppen Datenauswahl, Modellierung und Ausgabe beschreiben. Die Menüleiste bietet zudem zusätzliche Optionen:

Tabelle 31 Saiku Modellierungsobjekte

Option	Beschreibung
	legt ein neue Saiku Dashboard an
	öffnet ein Saiku Dashboard
	speichert ein Saiku Dashboard
	führt eine Abfrage manuell aus
	Schalter für automatisches Ausführen der Abfrage
	aus-/einblenden des Modellierungsbereichs
	aus-/einblenden der Datenauswahl
	gruppiert Parent Hierarchie Ebene
	aus-/einblenden von NULL Werten
	Spalten und Zeilen vertauschen
	Abfrage als MDX anzeigen
	führt Drill-Through auf markierter Zeile durch
	wie Drill-Through, aber mit Ausgabe als .csv Datei
	Export als .xlsx Datei
	Export als .csv Datei
	Export als .pdf Datei

	Wechsel zum MDX Editor
	ermöglicht markierte Felder mit Tags zu versetzen

6.3.1 Datenauswahl

Zur Datenauswahl ist es zunächst nötig, dass ein OLAP Schema veröffentlicht ist. Das Schema, bzw. der Cube ist anschließend in der Drop-Down-Liste verfügbar. Die Auflistung erfolgt dabei nach folgendem Schema:

Schema 1

Cube 1

Cube 2

Schema n

Cube m

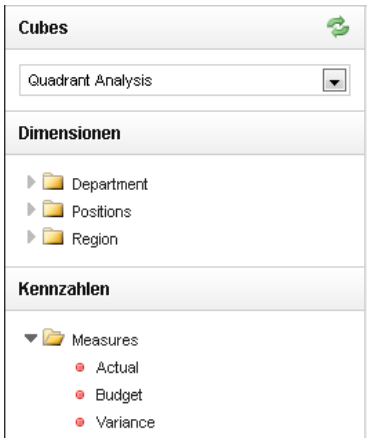


Abbildung 75 Saiku Datenauswahl

Nach Auswahl des Cubes werden automatisch die Dimensionen und Kennzahlen aufgelistet. Durch aufklappen der Dimensionen lassen sich die Hierarchien und ihre Ebenen anzeigen.

6.3.2 Modellierung

Aus der Datenauswahl lassen sich anschließend per Drag&Drop Elemente des Cubes in den Modellierungsbereich ziehen. Eine kontextabhängige Modellierungsassistenz zeigt dabei Modellierungsfehler auf. So können keine Ebenen derselben Hierarchie in Zeilen und Spalten gesetzt werden. Des Weiteren werden Dimensionen Blau, Kennzahlen Rot hinterlegt.

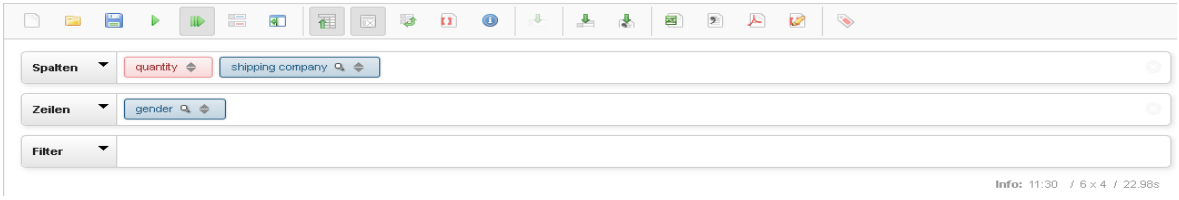


Abbildung 76 Saiku Modellierungsbereich

Die modellierte Abfrage wird standardmäßig direkt automatisch ausgeführt. Bei komplexeren Abfragen ist es zu empfehlen dieses zu deaktivieren. Im Hintergrund wird aus der Modellierung automatisch ein MDX Statement generiert, alternativ kann dieses auch manuell geschrieben werden.

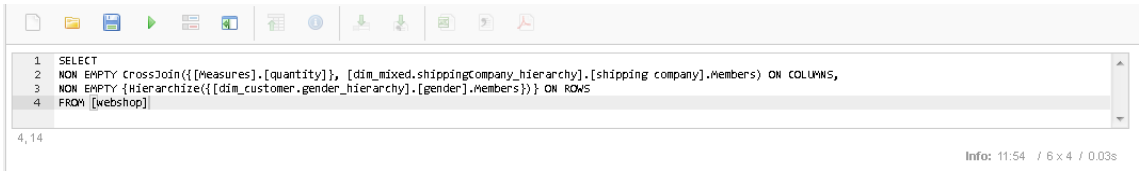

1 SELECT
2 NON EMPTY CrossJoin({[Measures].[quantity]}, [dim_mixed.shippingCompany_hierarchy].[shipping company].Members) ON COLUMNS,
3 NON EMPTY {Hierarchize({[dim_customer.gender_hierarchy].[gender].Members})} ON ROWS
4 FROM [webshop]
4, 14
Info: 11:54 / 6 x 4 / 0.03s

Abbildung 77 Saiku MDX Editor

Die im Modellierungsbereich abgelegten Symbole haben neben dem Bezeichner noch weitere Funktionen. Mit Klick auf die Lupe lassen sich einzelne Elemente der Ebene auswählen und mittels der Pfeile lässt sich die Ebene sortieren.

Abbildung 78 Saiku Hierarchieelement

Neben diesem Weg bietet Saiku weitere Wege zum Filtern und Sortieren der Abfrage. Zunächst lassen sich alle verwendeten Hierarchieebenen über das Lupen Symbol auf einzelne Elemente beschränken.

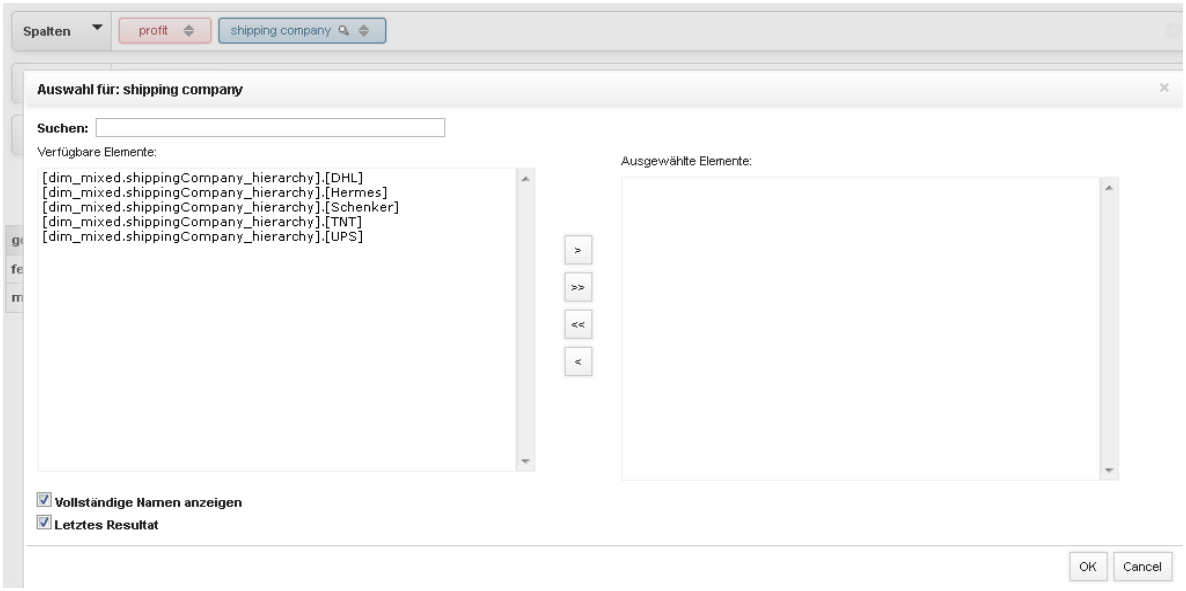


Abbildung 79 Saiku Elementfilter

Des Weiteren ist es möglich, eine Hierarchieebene als Filter zu verwenden, und diese auf einzelne Elemente zu beschränken. Zusätzlich können für die drei Modellierungsbereiche Spalte/Zeile/Filter jeweils noch übergreifende Filter, Limits und Sortierungen definiert werden. Bei allen drei Einstellungen lassen sich mittels MDX eigene Manipulationen erstellen. Limit bietet entweder vordefinierte Top/Bottom Filter, oder selbstdefinierte Top-/BottomCount/Percent/Sum Funktionen. Die Sort-Funktionen ermöglichen es, die Spalten/Zeilen nach den vorhandenen Kennzahlen zu sortieren, auch nach denen die nicht in der Ausgabe sind.

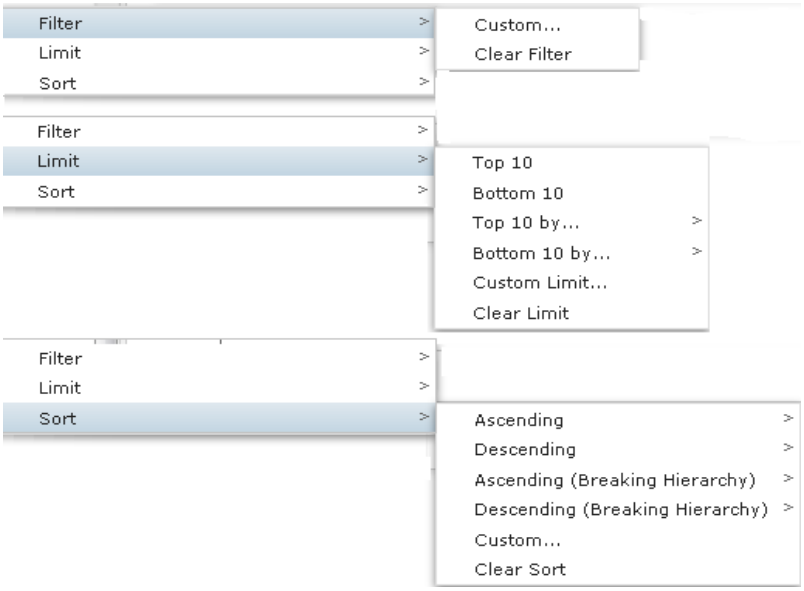


Abbildung 80 Saiku Filter/Limit/Sort Optionen

Die eigene Definition eines Filters ist ohne MDX Kenntnisse nicht möglich. Beim selbstdefinierten Filter wird der Achseninhalt in das Set_Expression Argument gelegt und die erstellte Expression im Filterdialog entspricht der Logical_Expression.[21]

Abbildung 81 Saiku Custom Filter

Widerspricht eine neue Filterung einer alten Filterung, so wird die alte verworfen und die neue beibehalten. Eine aktive Funktion wird durch eine Veränderung der Schrift von Normal auf Fett dargestellt.

6.3.3 Ausgabe

Das Ergebnis der Abfrage wird direkt als Tabelle ausgegeben, oder durch Wechsel des Modus in ein Diagramm umgewandelt.

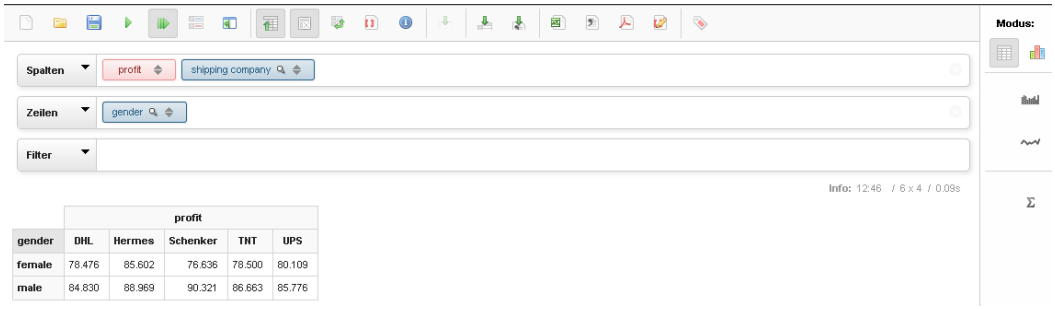


Abbildung 82 Saiku Ausgabe

[21] nach Syntax von:http://technet.microsoft.com/de-de/library/ms146037.aspx

Neben dem Abfrageergebnis werden zusätzlich Informationen zur Durchführung ausgegeben (Uhrzeit,Spalten-und Zeilenanzahl,Durchführungsdauer).

Der Diagrammmodus bietet verschiedene Diagramme zur Visualisierung der Abfrage. Die möglichen Diagramme sind fest vorgegeben und lassen sich nicht erweitern. Die Konfigurierbarkeit der Ausgabe ist stark beschränkt. Es ist möglich, Elemente der Spaltendimension in der Legende mit zu entfernen oder alle Zeilen eines Spaltenelements hervorzuheben. Mittels Mouse-over lassen sich Spalten-/Zeilenelement und Value des grafischen Elements anzeigen.

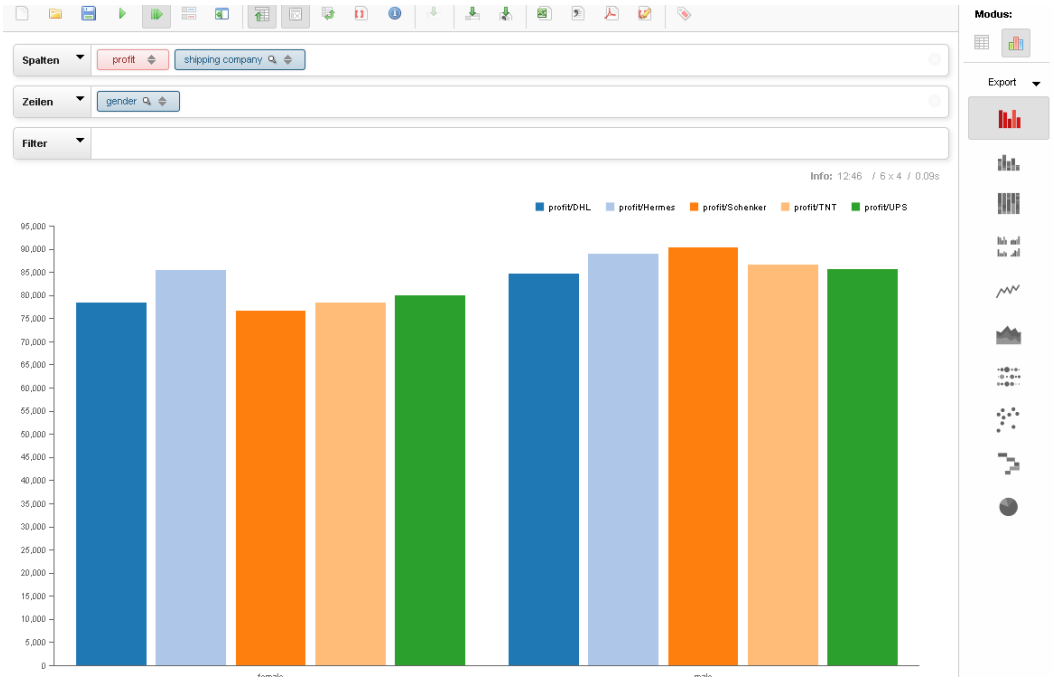


Abbildung 83 Saiku Diagrammausgabe

Wie schon in der Menüleiste beschrieben lassen sich Saiku Dashboards als .csv, .xlsx und .pdf exportieren. Im Diagrammmodus lassen sich diese zusätzlich in den Bildformaten .png, .jpeg, .svg exportieren. Des Weiteren lassen sich die Dashboards in den vorhandenen Ordnerstrukturen ablegen.

6.4 Dokumentation

Das Saiku Plugin hat keine Hilfe im Tool. Die Dokumentation liegt nur in englischer Sprache auf: http://docs.analytical-labs.com/ vor. Auf dem Entwickler Blog[22] werden zudem Patchnotes und eine Roadmap für künftige Versionen präsentiert. Das Forum[23] wird in einer Art Answerboard wie Yahoo's Ask.com geführt. Die Fragen werden mit Tags gelistet dargestellt und Antworten werden Bewertet und sortiert angeführt. Somit ist dies eher unstrukturiert und unübersichtlich. Derzeit ist zudem die Mehrheit der aufgeführten Fragen unbeantwortet. Die bisherigen Antworten sind nahezu vollständig unbewertet. Somit ist kein wirklicher Nutzen des Forums erkennbar.

6.5 Umsetzung Szenario

Anhand von vier Szenarios sollen die Grundfunktionen ausgetestet werden. Zunächst werden Filter, Sortierungs und Limitierungsfunktionen getestet, anschließend erfolgt die Nachbildung der PeriodsToDate Funktion über den Modeller, sowie über den MDX Editor. In allen Szenarien soll das Verhalten des Profits aller Bundesstaaten untersucht werden. Als Ausgangssituation werden alle Bundesstaaten und ihr Profit aller Jahre aufgelistet und das Verhalten mit einer Sparkline visualisiert. Es werden, falls nicht anders beschrieben, nur Funktionen des Modellers benutzt.

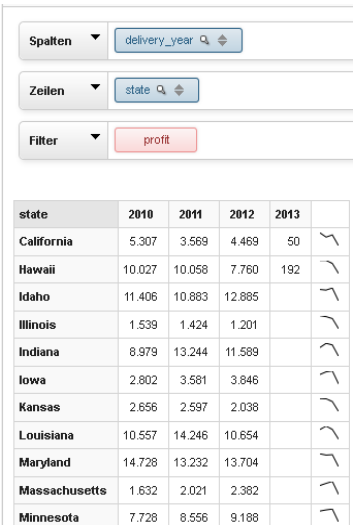


Abbildung 84 Saiku Szenario Ausgangsituation

[22] Siehe: http://blog.analytical-labs.com/
[23] Siehe: http://ask.analytical-labs.com/

6.5.1 Szenario 1 Filter

Zunächst erkennt man das für das Jahr 2013 nur sehr wenige Werte vorhanden sind und dadurch die Interpretation der Trends mittels Sparkline erschwert wird. Ziel ist es durch die verschiedenen Filterarten die Interpretation der Trends zu erleichtern. Um die Profittrends besser interpretieren zu können, wird zunächst die Hierarchieebene delivery_year auf die Elemente 2010,2011 und 2012 beschränkt. Somit fallen die größtenteils leeren Werte aus 2013 weg.

Abbildung 85 Saiku Szenario 1 Elementfilter

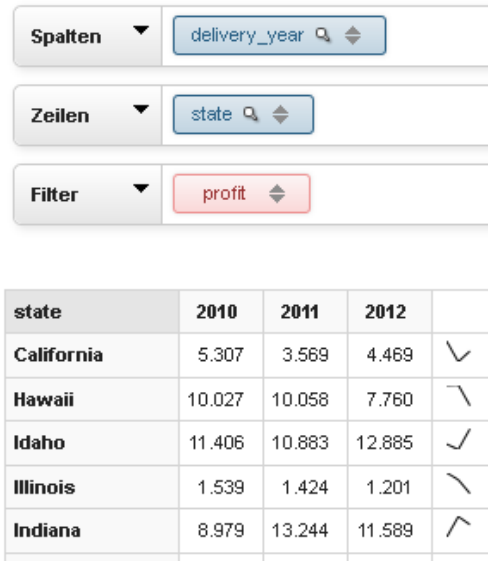


Abbildung 86 Saiku Szenario 1 Ausgabe Elementfilter

Bei der Visualisierung wird deutlich, dass die hohen Werte aus Tennessee die Interpretierung der weiteren Bundesstaaten erschweren. Grund ist der einheitliche Maßstab aller Diagramme.

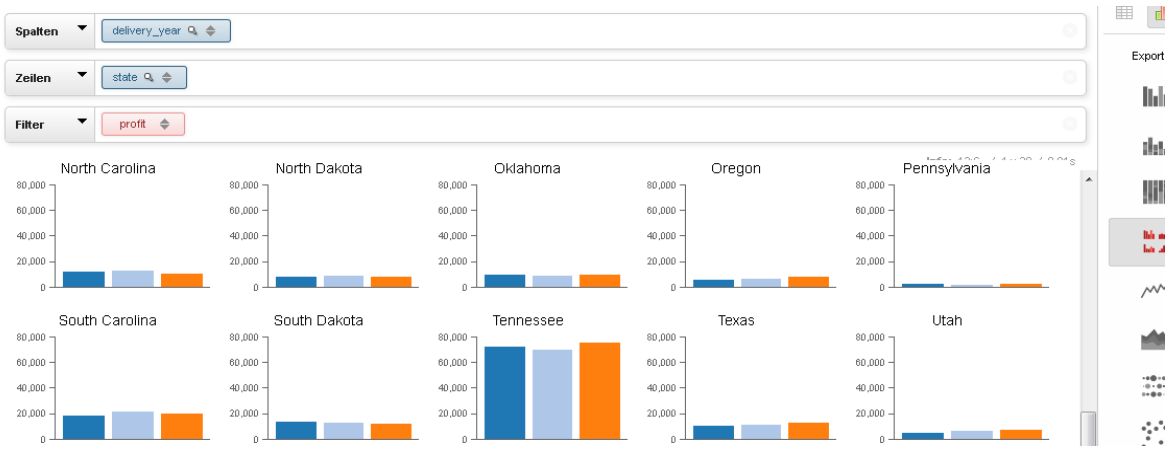


Abbildung 87 Saiku Szenario 1 grafische Ausgabe Elementfilter

Um die Bundesstaaten mit “normalen” Profit zu betrachten, wird ein Zeilenfilter angelegt, der nur noch die Bundesstaaten ausgibt, die einen Gesamtprofit (über alle Jahre) von weniger als 40.000 haben.

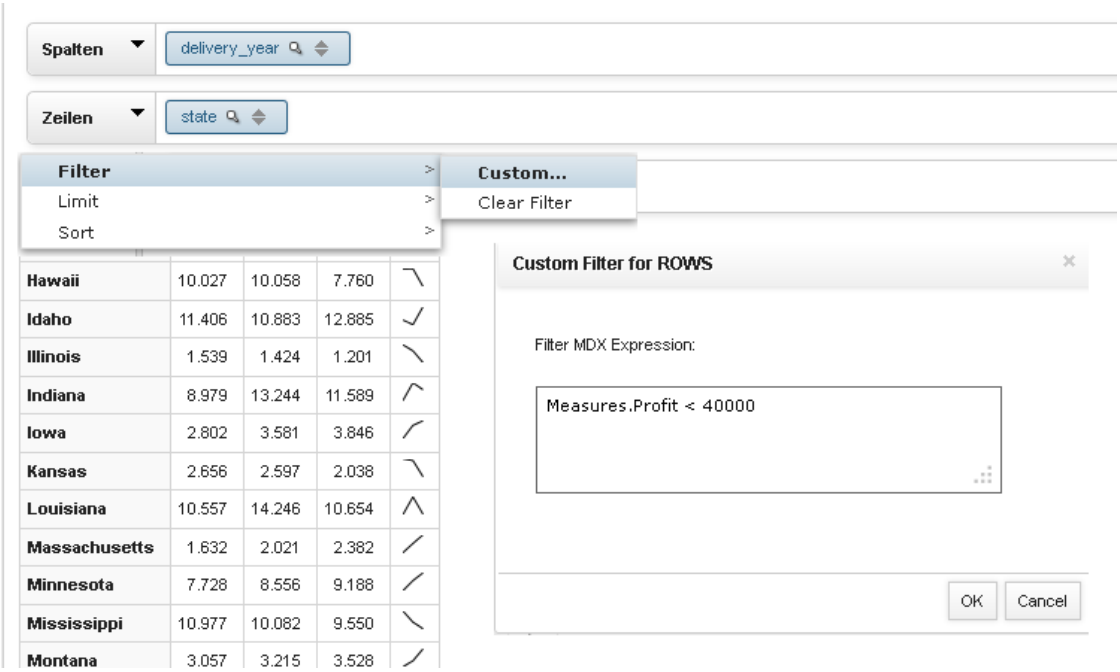


Abbildung 88 Saiku Szenario 1 Customfilter

Die Bundesstaaten Maryland, South Carolina und Tennessee werden somit gefiltert. Die Visualisierung ergibt nun besser interpretierbare Diagramme. In Louisiana sind beispielsweise gleich starke Schwankungen von und nach 2011 zu erkennen, während in Minnesota ein leichtes stetiges Wachstum erkennbar ist.

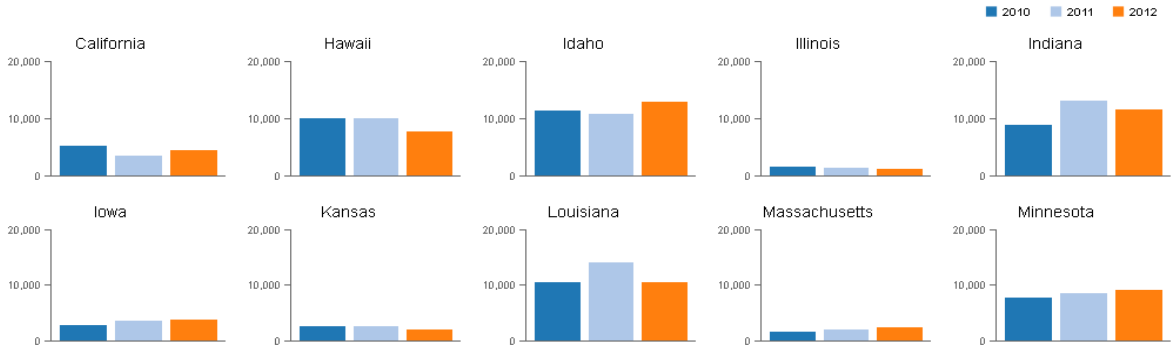


Abbildung 89 Saiku Szenario 1 Ausgabe Customfilter

Eine weitere Möglichkeit die Daten zu Filtern, ist die in die Profitsumme einbezogenen Daten zu verringern. Hier wird exemplarisch nur der Kundentyp “Business” betrachtet. Durch die Beschränkung auf den Kundentyp “Business” erscheinen die zuvor gefilterten Bundesstaaten Maryland und South Carolina wieder in den Auswertungen. Der zuvor gesetzte Filter auf den Gesamtprofit ist noch aktiv, jedoch ist die jetzige Profitsumme nur noch eine Teilmenge der Gesamtheit. In Louisiana ist zu erkennen, dass der Profit des Kundentyps “Business” stetig sinkt. In Anbetracht der zuvor beobachteten Schwankungen kann dies bedeuten das verstärkt Privatkunden in Louisiana Profit generieren.

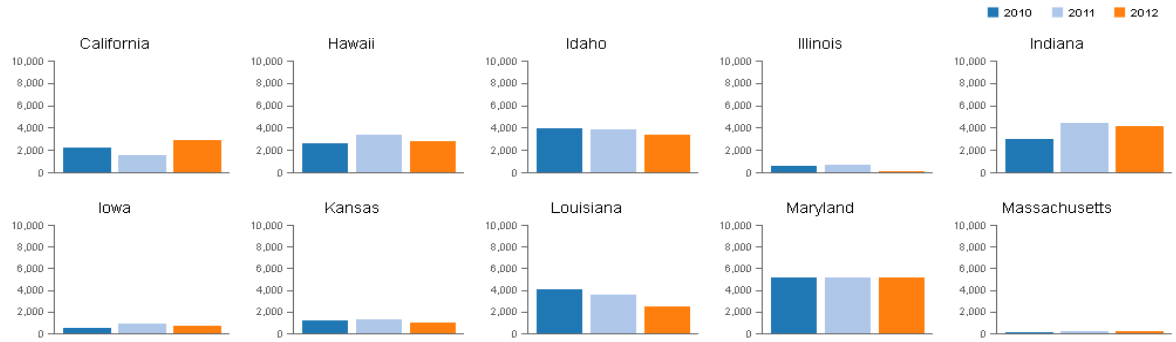


Abbildung 90 Saiku Szenario 1 Ausgabe Filter Kundentyp

6.5.2 Szenario 2 Sort

Standardmäßig werden die Werte alphabetisch Sortiert. Die Richtung kann wie schon im Kapitel 6.3 über die Pfeile am Icon gesteuert werden. Alternativ kann jede Achse nach allen verfügbaren Kennzahlen sortiert werden, auch jenen die nicht in der Auswertung enthalten sind. Für dieses Szenario werden die vorherigen zwei Filter (Profit<40000 und Type = Business) entfernt. In diesem Szenario wird die Ausgabe zunächst absteigend nach dem Profit sortiert, um schnell erfassen zu können, welcher Bundesstaat die höchsten Profite hat.

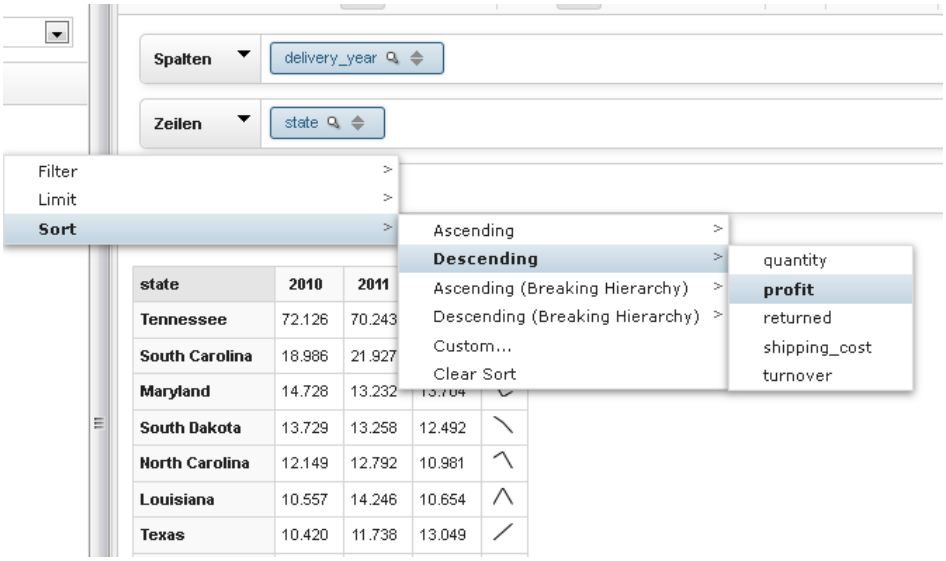


Abbildung 91 Saiku Szenario 2 Sort

In der Visualisierung wird deutlich, dass die Bundesstaaten Tennessee, South Carolina, Maryland und South Dakota die höchsten Profite haben.

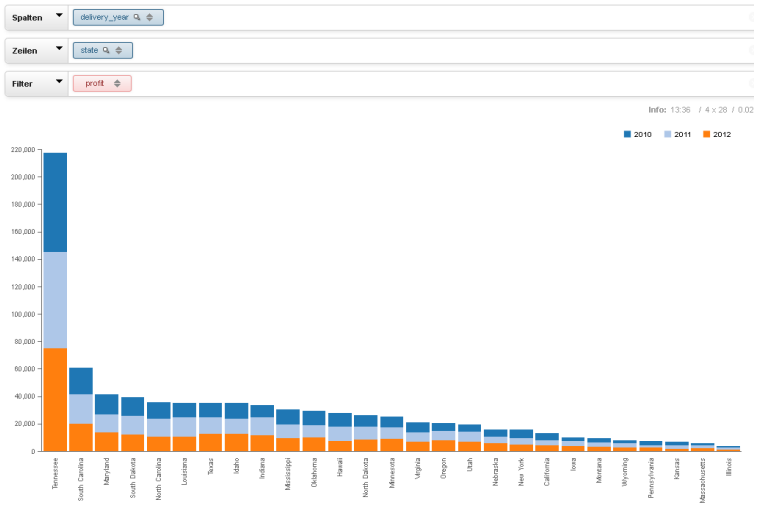


Abbildung 92 Saiku Szenario 2 Ausgabe

Durch den Weg der Datenerzeugung und Berechnung der Kennzahlen sind keine Änderungen bei Sortierungen nach anderen Kennzahlen zu erwarten.

6.5.3 Szenario 3 Limit

Für die Limit Funktionen(Top/Bottom) soll sich zunächst auf das Jahr 2012 beschränkt werden. Die vorgegebenen Top/Bottom 10 Funktionen entsprechen den MDX Funktionen Top/BottomCount und geben somit die ersten/letzten 10 Werte wieder. Top/Bottom 10 by… ergeben die besten/schlechtesten 10 Werte nach beliebiger Kennzahl wieder. Zur Ermittlung der Bundesstaaten, die die oberen 50 Prozent des Profits ausmachen wird ein Custom Limit angelegt.

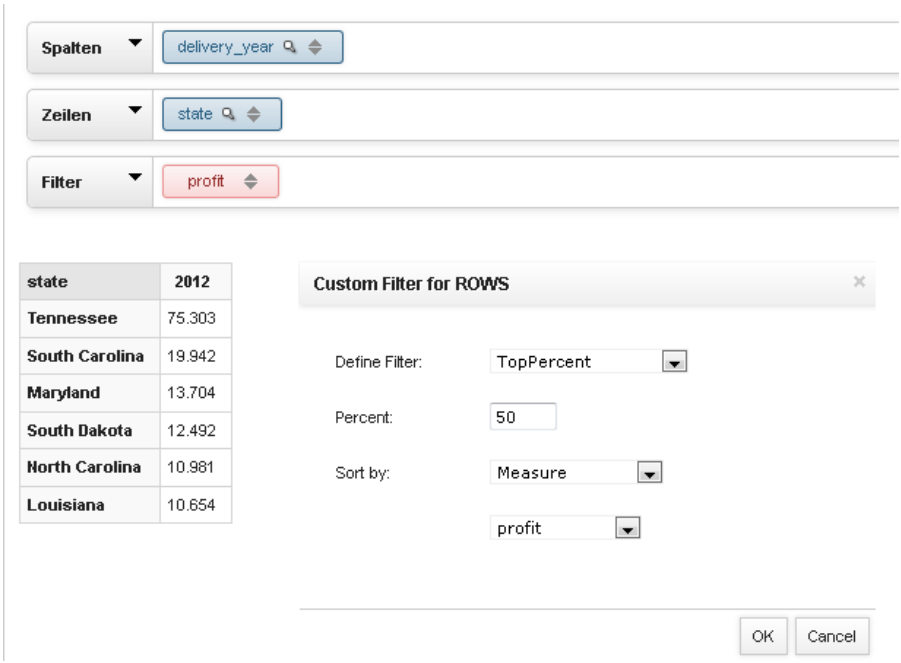


Abbildung 93 Saiku Szenario 3 Limit Ausgabe

Somit ist erkennbar, dass bloß 6 von 28 Bundesstaaten 50 Prozent des Profits erzeugen.

6.5.4 Szenario 4 PeriodsToDate

Mittels der PeriodsToDate Funktionen sollen die Profite der Bundesstaaten, der jeweils ersten sechs Monate verglichen werden. Zunächst wird dies mittels MDX Editor realisiert. Dazu wird pro Jahr jeweils ein Calculated Member erstellt. Diese benutzen die Periods-ToDate Funktion. Start der Periode ist die Jahresebene, Laufzeit der Periode ist jeweils das 6. Element der konkreten Jahresebene (2010/2011/2012).

```
1  with
2      member dim_delivery_date.Halbjahr1_2010 as aggregate(
3  PeriodsToDate(
4  dim_delivery_date.delivery_year, dim_delivery_date.[2010].[6]
5  ))
6
7      member dim_delivery_date.Halbjahr1_2011 as aggregate(
8  PeriodsToDate(
9  dim_delivery_date.delivery_year, dim_delivery_date.[2011].[6]
10 ))
11
12     member dim_delivery_date.Halbjahr1_2012 as aggregate(
13 PeriodsToDate(
14 dim_delivery_date.delivery_year, dim_delivery_date.[2012].[6]
15 ))
16
17 SELECT
18 NON EMPTY {dim_delivery_date.Halbjahr1_2010,dim_delivery_date.Halbjahr1_2011,dim_delivery_date.Halbjahr1_2012}
19 NON EMPTY {Hierarchize({[dim_geo].[state].Members})} ON ROWS
20 FROM [webshop]
21 where {[Measures].[profit]}
```

21, 27

state	Halbjahr1_2010	Halbjahr1_2011	Halbjahr1_2012
California	2.090	1.388	1.837
Hawaii	5.092	4.982	5.013
Idaho	6.109	5.158	6.572
Illinois	778	656	497
Indiana	4.156	6.064	6.117

Abbildung 94 Saiku Szenario 4 PeriodsToDate MDX

Die Umsetzung mittels Modeller Funktionen ist nicht möglich. Man kann zwar die einzelnen Monat betrachten, jedoch findet die Aggregation stets nur auf “All” Level statt.

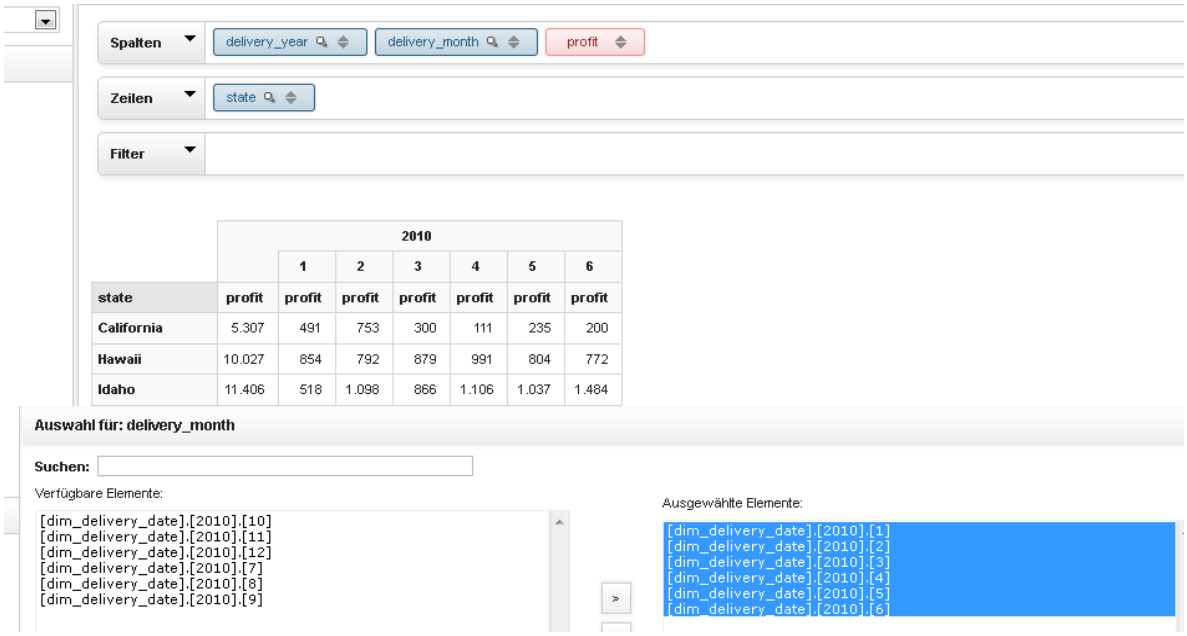


Abbildung 95 Saiku Szenario 4 Modeller

Generell lassen sich mittels des Modellers keine Calculated Members. Für die nächste Hauptversion (3.0) ist eine Unterstützung der Periodenfunktionen im Modeller angekündigt.[24]

[24]Siehe: http://blog.analytical-labs.com/post/61406811926/the-road-to-saiku-3

7 Data Mining

7.1 Übersicht

Data-Mining ist die systematische Anwendung mathematischer und statistischer Methoden auf einen Datenbestand, mit dem Ziel neue Muster zu erkennen, die vorzugsweise in Data Warehouses oder Data Marts abgelegt sind. Üblicherweise können diese Muster nicht durch das klassische Durchsuchen von Daten ermittelt werden, da die Beziehungen zu komplex sind, bzw. zu viele Daten bestehen.

Für die Analyse der Daten werden allgemeine und effiziente Methoden bereitgestellt. Die Systeme sollen dabei in der Lage sein, die geeigneten Methoden selbstständig auszuwählen, die Daten zu analysieren und zu visualisieren. Als gängige Verfahren des Data-Minings sind Clustering, Entscheidungsbäume, Assoziationsanalysen und Neuronale Netze zu nennen.

In Pentaho werden viele BI Komponenten zwar unterstützt, jedoch wurden für die Data-Mining Komponente keine Funktionen integriert. Diese wird im Pentaho über die Schnittstelle zur statistischen Software Weka ermöglicht. Die Data-Mining Komponente ging aus dem Projekt, Weka, der neuseeländischen Universität Waikato hervor.

7.2 Funktionalität

Weka ist als Data-Mining Tool besonders geeignet für die Erstellung von Cluster-, Klassifikations- und Assoziationsanalysen, sowie für die Erstellung von Entscheidungsbäumen. Anhand eines Plugins können in Pentaho alle in Weka integrierten Methoden und Funktionen des Data-Minings verwendet werden.

Der Zugriff auf Weka über Pentaho ist anhand des Weka Scoring Plugins möglich. Das Plugin ist bereits in Pentaho integriert und wird bei Bedarf durch die Option Marketplace, in dem Menüpunkt „Help“, der Software installiert bzw. eingerichtet.

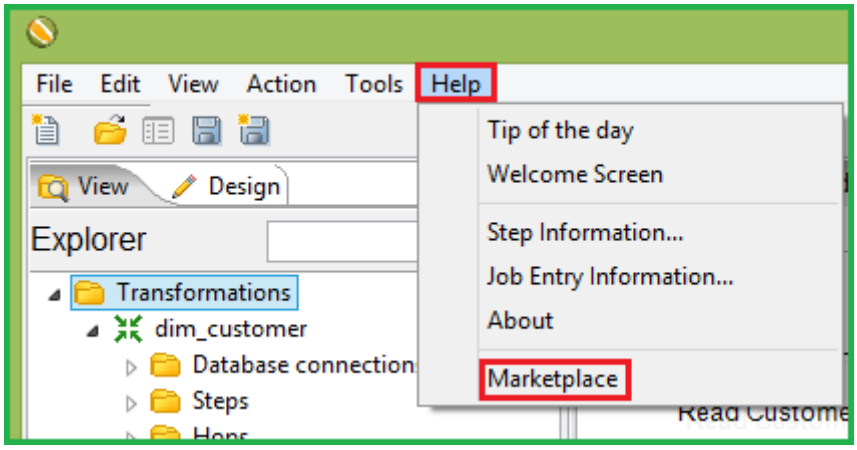


Abbildung 96 PDI Marketplace

Das Installieren des Weka Scoring Plugins wird in der Community Edition von Pentaho leider nicht unterstützt, daher ist das Data-Mining in der Community Edition nicht möglich jedoch sind in der Enterprise Edition von Pentaho alle Funktionen des Data-Minings verfügbar. Plugins werden im Menü Help auf den Punkt Marketplace aufgelistet.

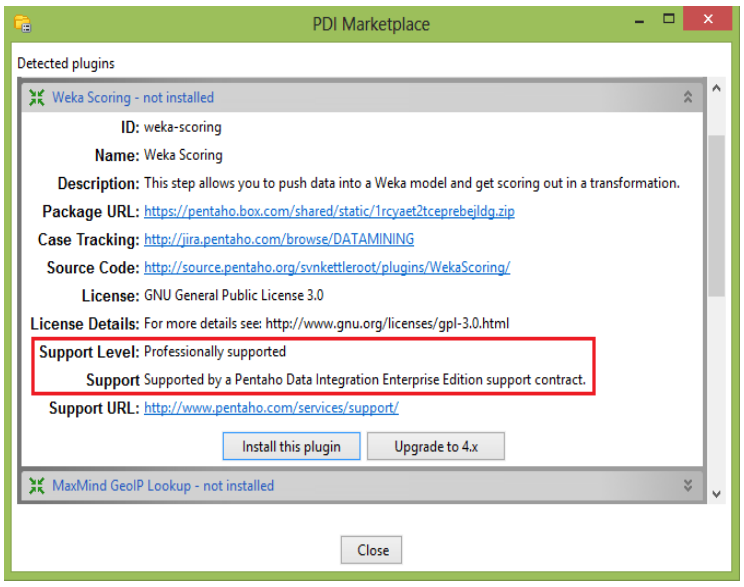

PDI Marketplace
Detected plugins
Weka Scoring - not installed
ID: weka-scoring
Name: Weka Scoring
Description: This step allows you to push data into a Weka model and get scoring out in a transformation.
Package URL: https://pentaho.box.com/shared/static/1rcyaet2tceprebejldg.zip
Case Tracking: http://jira.pentaho.com/browse/DATAMINING
Source Code: http://source.pentaho.org/svnkettleroot/plugins/WekaScoring/
License: GNU General Public License 3.0
License Details: For more details see: http://www.gnu.org/licenses/gpl-3.0.html
Support Level: Professionally supported
Support Supported by a Pentaho Data Integration Enterprise Edition support contract.
Support URL: http://www.pentaho.com/services/support/
Install this plugin
Upgrade to 4.x
MaxMind GeoIP Lookup - not installed
Close

Abbildung 97 PDI Weka Scoring Plugin Eigenschaften

Die Abbildung 97 PDI Weka Scoring Plugin Eigenschaften zeigt deutlich an, dass die Aktivierung des Weka's Plugin für eine Community Edition von Pentaho nicht möglich ist. Selbst beim Versuch, dieses Plugin für eine CE zu installieren wird eine Fehlermeldung angezeigt und der Vorgang ist abzubrechen.

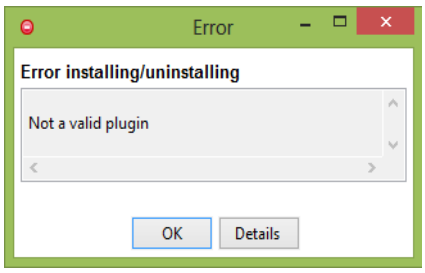


Abbildung 98 PDI Weka Fehlermeldung

7.3 Dokumentation

Pentaho bietet im Wiki für Data-Mining und explizit dem Weka Plugin Dokumentationen an.[25] Die zur Verfügung gestellten Dokumentationen sind hauptsächlich Lernmaterialien und werden in der Pentaho-Wiki zusammengefasst.

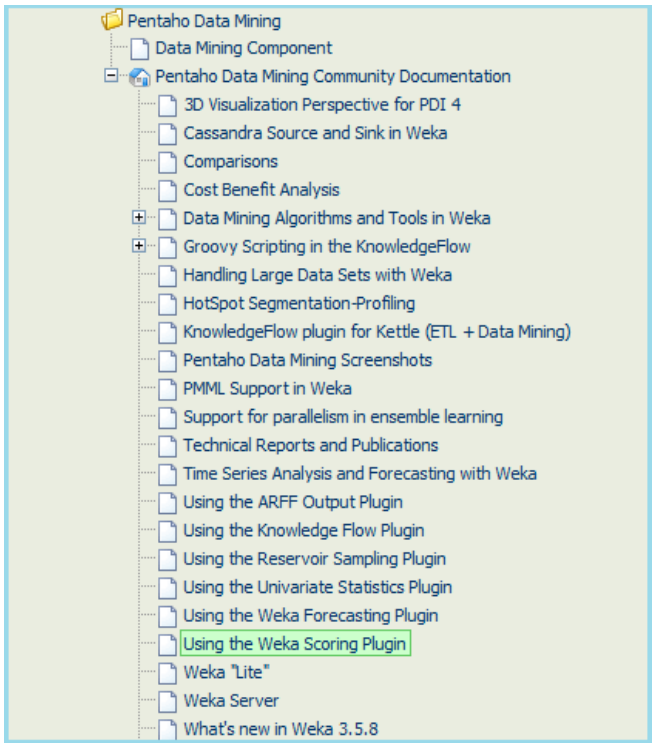

Pentaho Data Mining
Data Mining Component
Pentaho Data Mining Community Documentation
3D Visualization Perspective for PDI 4
Cassandra Source and Sink in Weka
Comparisons
Cost Benefit Analysis
Data Mining Algorithms and Tools in Weka
Groovy Scripting in the KnowledgeFlow
Handling Large Data Sets with Weka
HotSpot Segmentation-Profiling
KnowledgeFlow plugin for Kettle (ETL + Data Mining)
Pentaho Data Mining Screenshots
PMML Support in Weka
Support for parallelism in ensemble learning
Technical Reports and Publications
Time Series Analysis and Forecasting with Weka
Using the ARFF Output Plugin
Using the Knowledge Flow Plugin
Using the Reservoir Sampling Plugin
Using the Univariate Statistics Plugin
Using the Weka Forecasting Plugin
Using the Weka Scoring Plugin
Weka "Lite"
Weka Server
What's new in Weka 3.5.8

Abbildung 99 Pentaho Wiki / Weka Scoring Plugin

[25] Siehe:
http://wiki.pentaho.com/display/DATAMINING/Pentaho+Data+Mining+Community+Documentation

Der Angebotskatalog zum Data Mining liefert erforderliche Informationen sowohl für die Einrichtung und Installation von Weka und vom Weka Scoring Plugin als auch für das Deployment des Weka Data Mining Modells und die Integration der Algorithmen in Pentaho.

Einführung in Weka (Pentaho Data Mining Weka)

Diese Dokumentation beschreibt alle Schritte, von der Installation bis zur korrekten Anwendung der Software Weka. Dieses Benutzerhandbuch ist als PDF-Datei herunterzuladen. Es enthält 327 Seiten und als Standardsprache ist Englisch vorgesehen. Dabei wird erklärt, wie das Data Mining in Weka funktioniert.
Algorithmen und Data Mining Modell und -Strukturen werden beschrieben und anhand einfacher Beispiele und Screenshots verdeutlicht. Kenntnisse über Data Mining setzen zwar eine schnelle und nachvollziehbare Verständnis dieser Dokumentation voraus, sind aber keine Voraussetzungen, um Data Mining Modell zu generieren. Das Handbuch ist auf dem Server im Installationspfad von Weka zu finden.

Installieren und Inbetriebnahme der Weka-Scoring Plugin in Pentaho

Zusätzlich bietet diese Dokumentation Hilfestellungen zur Installation und Benutzung des Weka Scoring Plugin in Pentaho an. Anhand von Screenshots und einfachen Anwendungsbeispielen wird dargestellt, wie das Plugin zu finden und zu installieren ist; wie die im Weka erstellte Miningmodelle im Pendaho Data Integration zu implementieren sind, und wie Data Mining in Pentaho (Enterprise Edition) zu realisieren ist.

Die Dokumentationen sind gut strukturiert ermöglichen eine schnelle Einführung in Data Mining mit Weka und der Umsetzung in mit Pentaho. Die Beispiele sind jedoch nicht sehr komplex, wodurch nicht die gesamten Funktionen und Fähigkeiten von Pentaho und Weka abgedeckt werden.

7.4 Umsetzung Szenario

Da das Weka Plugin nicht in der Community Edition nutzbar ist, wird in diesem Kapitel lediglich in der Theorie beschrieben, welche Schritte nötig sind um dies in der Enterprise Edition umsetzen zu können.

Einrichtung der Data-Mining Komponente in der Enterprise Edition:

Wenngleich die Data-Mining Komponente von der Community Edition nicht unterstützt wird, sind zwei Hauptregeln für eine reibungslose Benutzung der Data-Mining Komponente in der professionellen Edition zu beachten.

- Die Software Weka muss mindestens ab der Version 3.5.7 auf dem Rechner installiert sein.
- Das Weka scoring Plugin der ETL-Software Pentaho Data Integration muss vorab eingerichtet sein.

Auf der Homepage von Pentaho kann das Weka Scoring Plugin ab der Version 3.6 heruntergeladen werden. Nach dem Entpacken der Zip-Datei sind alle Komponenten des WekaScoringDeploy, in dem Ordner, in dem PDI installiert wurde, zu installieren. Schließlich wird die "weka.jar" Datei vom Weka 3.7.5 zum gleichen Ordner hinzugefügt.[26]

In Weka muss zunächst ein Data-Mining Modell (Clustermodell, Assoziationsmodell, Klassifikationsmodell, Entscheidungsbaum-Modell usw.) erstellt werden. Dieses Modell wird im nächsten Schritt in Pentaho exportiert, dort wird es als Vorlage für die Erstellung eines weiteren Mining Modells verwendet.

Die Data-Mining Algorithmen Weka`s können in Pentaho nicht direkt verwendet werden. Erst nach ihrer Erstellung in Weka kann das Modell exportiert werden. Bedingung dafür ist eine erfolgreiche Installation des Weka scoring Plugins; was In der Community Edition nicht möglich ist.

Nach dem Export des Modells können nun, in der Pentaho Software, weitere Mining Modelle erstellt werden. Dabei wird nur die Quelldatei (csv, excell usw...) aktualisiert, bzw. geändert und das ganze Modell muss noch einmal eingespielt werden, da das Weka Modell als Vorlage verwendet wird.

[26] Server Dateipfad: ...\pdi-ce-4.4.0-stable\data-integration\plugins\steps

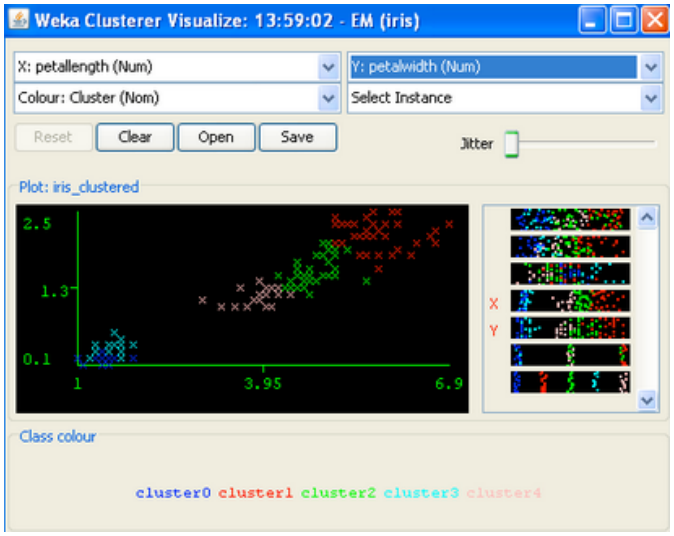


Abbildung 100 Weka Clustering Ausgabe

Als Beispiel wird hier ein Clustering der Kundendaten durchgeführt. Die Datei wird unter dem Namen “Cluster_model_customer” auf dem Server im Ordner Weka gespeichert. Dieses Clustermodell kann nun im Pentaho Data Integration über das Plugin eingelesen. Zuerst muss eine Transformation erstellt werden. Dabei erfolgt das Einlesen des Weka’s Modells über das Modellierungsobjekt „Weka Scoring”.

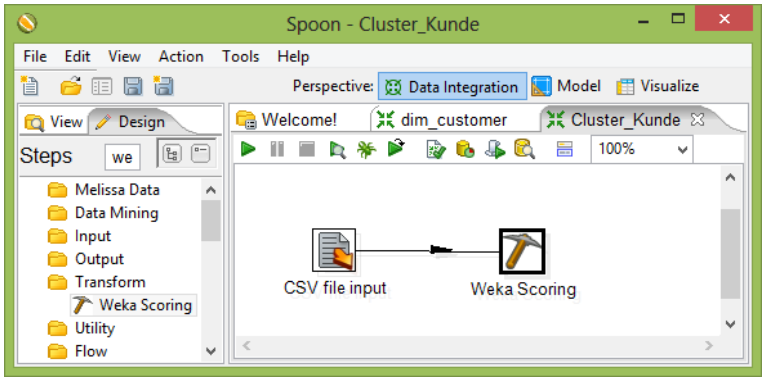


Abbildung 101 PDI Transformation Clustering

In den Reitern “Fields mapping” und “Model” ist das Cluster-Modell und die Darstellung exakt übernommen worden, genau wie von Weka erstellt. Schließlich ist das Modell auszuführen und die erstellten Cluster anzusehen. Kunden werden nach den von Weka gewünschten angebotenen Cluster Methoden (K-Means, EM, Hierarchical Clustering usw.), Cluster-Parametern und gewünschten Clusteranzahl gruppiert.

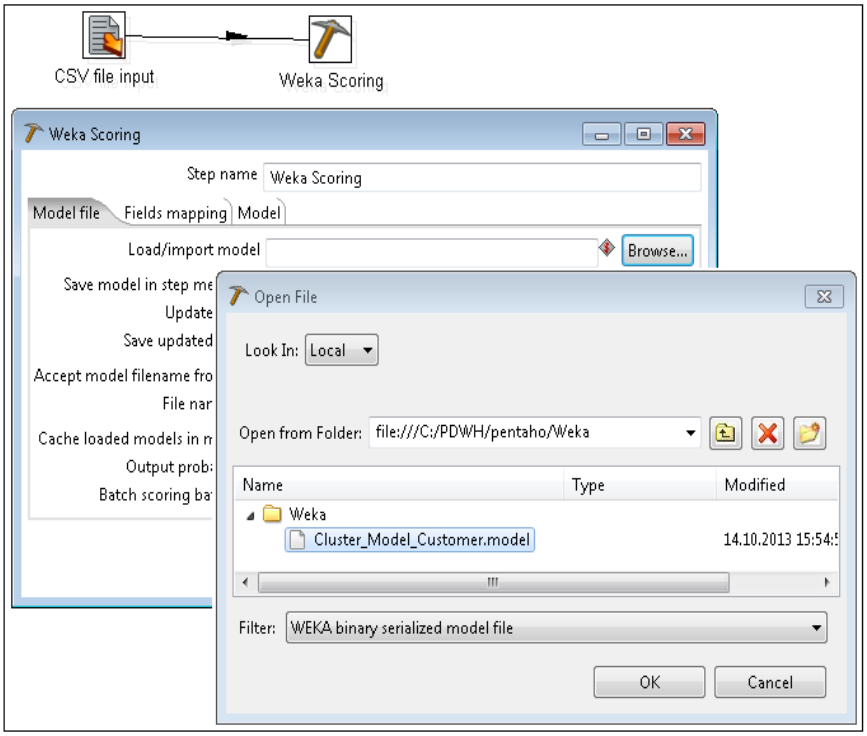


Abbildung 102 PDI Weka Scoring Objekt

PDI gibt das Ergebnis dieses Clustering anschließend als Tabelle aus.

input15	input16	class	class:0_predicted_prob	class:1_predicted_prob	class:2_predicted_prob	class:3_predicted_prob	class:4_p
040	098	8	0.002	0	0	0	
100	006	2	0	0.0096	0.9904	0	
016	000	1	0	0.9981	0	0	
067	000	4	0	0	0	0	
047	000	1	0	0.9412	0.0588	0	
020	020	6	0	0	0	0	
066	000	4	0	0	0	0	
000	057	0	0.9961	0	0	0	
100	100	5	0	0	0	0	
026	085	0	0.9961	0	0	0	
018	001	9	0	0	0	0.0065	
000	085	8	0.002	0	0	0	
000	002	5	0	0	0	0	
021	000	9	0	0	0	0	
100	024	7	0	0	0	0	
000	017	3	0	0	0	1	
000	007	3	0	0.0034	0	0.992	
000	016	9	0	0	0	0	

Abbildung 103 PDI Weka Scoring Ausgabe

8 Fazit

Im Folgenden werden die einzelnen Komponenten der Pentaho Business Analytics Community Edition im Kontext der Erfahrungen aus der Evaluation abschließend besprochen. Am Ende wird ein Gesamtfazit bezüglich des Einsatzes in der Lehre gezogen.

Der BI-Server, als auf Apache Tomcat basierender Application-Server, stellt in der Community Edition lediglich Basisfunktionen bereit. Dazu gehören rudimentäre administrative Fähigkeiten in Form der Administrator Console sowie Frontendfunktionalität mittels der User Console. Weiterhin enthält er die OLAP-Erweiterung Pentaho Mondrian als Basis für multidimensionale Auswertungen. Über den Pentaho Marketplace werden verschiedene Erweiterungsmöglichkeiten für den Endanwender bereitgestellt. Der BI-Server lief während der Evaluationsphase stabil. Durch die Client-Server Architektur und die webbasierten Oberflächen kann die Funktionalität einfach zur Verfügung gestellt werden. Während der Server zentral durch die HTW administriert wird, könnten die Studenten mit den Client-Tools oder per Webbrowser über das Netzwerk darauf zugreifen.

Pentaho Data Integration, kurz PDI, bietet als eigenständiges Tool wesentliche Funktionen für die Umsetzung eines ETL-Prozesses. Es konnten sämtliche für die Erzeugung des Star-Schemas notwendigen Transformationen umgesetzt werden. Auch die Möglichkeiten zur Fehlerbehandlung und zum Debugging können überzeugen und stehen kommerziellen Tools in nichts nach. Insbesondere die Möglichkeit eigenen Java-Quellcode einbinden zu können und die vordefinierten Schnittstellen-Objekte zu anderen Tools, bietet eine hohe Flexibilität. Insgesamt macht die Anwendung einen ausgereiften Eindruck und es lassen sich in kurzer Zeit brauchbare Ergebnisse erzeugen. Lediglich die Überschneidung in der Funktionalität vieler Transformationsobjekte fällt leicht negativ auf. Hier könnte eine Konsolidierung vonseiten Pentaho Sinn machen.

Pentaho Mondrian und die Schema Workbench decken das Thema (R)OLAP sowohl von der Funktionalität als auch vonseiten der Modellierung umfassend ab. Mondrian ist direkt in den BI-Server integriert und muss standardmäßig nicht weiter konfiguriert oder administriert werden. Durch den Wegfall einer weiteren multidimensionalen Datenbank bzw. der direkten Nutzung des relationalen Modells wird die Komplexität reduziert. Die Schema Workbench macht einen etwas veralteten Eindruck, z. B. erkennt der integrierte Validator keine degenerierten Dimensionen, obwohl diese explizit Bestandteil der Funktionalität von Mondrian sind. Aufgrund fehlender Assistenten oder integrierter Hilfefunktionen gestaltet sich das Modellieren mühsam.

Ohne ein grundlegendes Verständnis der zugrunde liegenden XML-Strukturen ist die Erstellung eines funktionierenden Modells nicht möglich. Dies erfordert Einarbeitungszeit vonseiten des Nutzers, ist aufgrund der vorhandenen Dokumentation jedoch möglich.

Der Report-Designer bietet statisches Reporting sowohl auf MDX als auch auf SQL-Datenquellen. Die Abfragen können manuell innerhalb des Report-Designers verfasst werden. Zusätzlich ist es möglich, SQL-Abfragen durch einen Abfrageassistenten unterstützt zu generieren. Das Reportlayout kann mittels Drag&Drop relativ einfach erzeugt werden, wofür verschiedene Templates bereitgestellt werden. Es lassen sich somit pixelgenaue Reports erstellen, die in verschiedenen Standardformaten (PDF, Excel, RTF, CSV) exportiert werden können. Das Datenmaterial kann durch Diagramme visualisiert werden und durch entsprechende Verwendung der vorhandenen Steuerelemente aufbereitet werden. Reports können nach kurzer Einarbeitungszeit schnell und einfach erstellt werden.

Saiku Analytics, als Add-On aus dem Pentaho-Marketplace, ermöglicht multidimensionale Auswertungen im Browser. Diese können Tabellarisch oder mittels Diagrammen dargestellt werden. Es bietet jedoch keine echte Dashboard-Funktionalität, da z. B. die interaktive Navigation auf den erstellten Auswertungen/Diagrammen nicht möglich ist. Weiterhin kann immer nur eine Abfrage als Basis verwendet werden. Während der MDX-Editor komplexe Abfragen unterstützt, können Abfragen auch mittels Drag&Drop relativ einfach erstellt werden. Dabei treten jedoch Beschränkungen auf (keine calculated members), sodass komplexere Abfragen teilweise manuell modifiziert werden müssen.
Durch die einfache und intuitive Bedienung können Abfragen ad hoc und mit sofortigem Ergebnis dargestellt werden. Es handelt sich somit eher um einen OLAP-Viewer mit Diagramm-Funktionalität. An dieser Stelle zeigt sich die fehlende Funktionalität der Community Edition auf dem Gebiet der Visualisierung bzw. der Dashboards.
Das Data Mining mittels Weka wurde aufgrund der Limitierungen in der Community Edition nur ohne Integration mit PDI durchgeführt. Es werden die Standardfunktionen des Data Mining (z. B. Clusteranalyse) unterstützt.

Zusammenfassend ergibt die Evaluation der Pentaho Business Analytics Community Edition ein gemischtes Bild: Durch den Einsatz von Standardtechnologien (Java, JDBC, XML, HTML, Javascript) und dem Open Source-Modell bieten die einzelnen Komponenten ein hohes Maß an Anpassbarkeit, Flexibilität und Kompatibilität.

Sollte sich eine verteilte Installation und Nutzung (BI-Server HTW, Client-Tools Studenten) aufgrund fehlender administrativer Funktionalität, insbesondere eines Rollen- und Zugriffsmodells, als nicht praktikabel erweisen, kann diese aufgrund der freien Verfügbarkeit aller Komponenten auch durch die Studierenden selbst erfolgen. Auch waren sowohl Server als auch Client-Tools während der Erprobung weitgehen stabil. Die Funktionalität im Bereich ETL und OLAP ist überzeugend und mit kommerziellen Tools vergleichbar. Dem Gegenüber steht die teilweise fehlende Funktionalität im Bereich Visualisierung und Dashboards.

Als Vor- und Nachteil zugleich kann die Modularität gesehen werden: Die inkonsistenten, Oberflächen und Bedienkonzepte sowie fragmentierten Dokumentationen bedeuten eine lange Einarbeitungszeit. Weiterhin ist zwischen den Komponenten eine manuelle Integration mittels Schnittstellen und/oder Dateien nötig. Auf der anderen Seite hat jedes Tool einen relativ klar abgegrenzten Aufgabenbereich und eigene Ergebnisartefakte. Damit können Teilbereiche wie z. B. OLAP abgegrenzt und ohne zusätzlichen funktionalen Overhead einer integrierten BI-Suite betrachtet werden. Die Komplexität des Gesamtthemas wird dadurch zerlegt und kann leichter vermittelt werden. Insbesondere in den Teilbereichen ETL und OLAP erscheint daher ein Einsatz zu Lehrzwecken in der HTW durchaus sinnvoll.

Anhang

A0 Arbeitsumgebung

Zugangsdaten des Servers

Remotedesktop
ip: 141.45.20.245
User: pentaho
Pass: d5h7#1xy

MySQL Credentials
User: root
Pass: password

Userconsole
url: http://localhost:8080/pentaho/
User: joe
Pass: password

Adminconsole
url: http://localhost:8099/
User: admin
Pass: password

Connectiondefinition

Hostname: localhost
Port: 3306
dbname: webshop
Driverclassname: com.mysql.jdbc.Driver
username: root
password: password
url: jdbc:mysql://localhost:3306/webshop

Änderungen an Dateien:

Definition des Publisher Passwort:
C:\PDWH\pentaho\biserver\pentaho-solutions\system\publisher_config.xml
<publisher-password>password</publisher-password>

Pfade der Arbeitsumgebung:

MySQL Server 5.6.11
C:\PDWH\MySQL\mysql-5.6.11-win32

MySQL Workbench 6.0
C:\PDWH\mysqlworkbench\MySQLWorkbench.exe

Java 32Bit
C:\Program Files\Java\jdk1.7.0_21
C:\Program Files\Java\jre7

Weka 3.6
C:\Program Files\Weka-3-6

User Console 4.8.0
C:\PDWH\pentaho\biserver\biserver-ce\start_pentaho.bat
C:\PDWH\pentaho\biserver\biserver-ce\stop_pentaho.bat

Admin Console 4.8.0
C:\PDWH\pentaho\biserver\administration-console\start_pac.bat
C:\PDWH\pentaho\biserver\administration-console\stop_pac.bat

PDI 4.4.0
C:\PDWH\pentaho\pdi-ce-4.4.0-stable\data-integration\Spoon.bat

PRD 3.9.1
C:\PDWH\pentaho\prd-ce-3.9.1\report-designer\report_designer.bat

PSW 3.5.0
C:\PDWH\pentaho\psw-ce-3.5.0\schema-workbench\workbench.bat

Ergebnisdateien

SQL Script
C:\PDWH\pentaho\SQL Script\create_script_3108.sql

Quelldaten (csv)
C:\PDWH\pentaho\Quelldaten

Transformationen
C:\PDWH\pentaho\transformationen\pdi_transformationen

Jobs
C:\PDWH\pentaho\Jobs

OLAP Schema
C:\PDWH\pentaho\olap

Reports
C:\PDWH\pentaho\Reports

Dashboards
C:\PDWH\pentaho\biserver\biserver-ce\pentaho-solutions\webshop

A1 SQL Create Script

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `webshop` DEFAULT CHARACTER SET utf8 ;
USE `webshop` ;

-- -----------------------------------------------------
-- Table `webshop`.`dim_customer`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `webshop`.`dim_customer` ;

CREATE  TABLE IF NOT EXISTS `webshop`.`dim_customer` (
  `CUSTOMER_ID` bigINT(11) NULL DEFAULT null ,
  `NAME` VARCHAR(255) NULL DEFAULT null ,
  `BIRTHDAY` DATETIME NULL DEFAULT null ,
  `GENDER` VARCHAR(255) NULL DEFAULT null ,
  `CUSTOMER_TYPE` VARCHAR(255) NULL DEFAULT null ,
  `Customer_scd_tk` bigint(20)not null,
  `version` int(11) DEFAULT null,
  `date_from` datetime DEFAULT null,
  `date_to` datetime DEFAULT null,
  `load_Date` datetime DEFAULT null,
  PRIMARY KEY (`Customer_scd_tk`)
 ,KEY `idx_dim_customer_scd_tk` (`Customer_scd_tk`)
)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;



-- -----------------------------------------------------
-- Table `webshop`.`dim_date`
```

```
-- -----------------------------------------------------
DROP TABLE IF EXISTS `webshop`.`dim_date` ;

CREATE  TABLE IF NOT EXISTS `webshop`.`dim_date` (
  `DATE_ID` INT(11) NOT NULL ,
  `DATE` DATETIME NULL DEFAULT NULL ,
  `DAY` INT(11) NULL DEFAULT NULL ,
  `MONTH` INT(11) NULL DEFAULT NULL ,
  `YEAR` INT(11) NULL DEFAULT NULL ,
  PRIMARY KEY (`DATE_ID`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `webshop`.`dim_geo`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `webshop`.`dim_geo` ;

CREATE  TABLE IF NOT EXISTS `webshop`.`dim_geo` (
  `GEO_ID` INT(11) NOT NULL ,
  `LAT` VARCHAR(255) NULL DEFAULT NULL ,
  `LON` VARCHAR(255) NULL DEFAULT NULL ,
  `ZIPCODE` VARCHAR(255) NULL DEFAULT NULL ,
  `CITY` VARCHAR(255) NULL DEFAULT NULL ,
  `STATE` VARCHAR(255) NULL DEFAULT NULL ,
  PRIMARY KEY (`GEO_ID`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `webshop`.`dim_mixed`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `webshop`.`dim_mixed` ;

CREATE  TABLE IF NOT EXISTS `webshop`.`dim_mixed` (
```

```
  `MIXED_ID` INT(11) NOT NULL ,
  `SHIPPING_COMPANY` VARCHAR(255) NULL DEFAULT NULL ,
  `PRODUCT_GROUP` VARCHAR(255) NULL DEFAULT NULL ,
  PRIMARY KEY (`MIXED_ID`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `webshop`.`dim_product`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `webshop`.`dim_product` ;

CREATE  TABLE IF NOT EXISTS `webshop`.`dim_product` (
  `PRODUCT_ID` INT(11) NOT NULL ,
  `NAME` VARCHAR(255) NULL DEFAULT NULL ,
  `PRODUCER` VARCHAR(255) NULL DEFAULT NULL ,
  PRIMARY KEY (`PRODUCT_ID`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `webshop`.`dim_time`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `webshop`.`dim_time` ;

CREATE  TABLE IF NOT EXISTS `webshop`.`dim_time` (
  `TIME_ID` INT(11) NOT NULL ,
  `HOUR` INT(11) NULL DEFAULT NULL ,
  `MINUTE` INT(11) NULL DEFAULT NULL ,
  PRIMARY KEY (`TIME_ID`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
```

```
-- Table `webshop`.`fact_delivery_item`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `webshop`.`fact_delivery_item` ;

CREATE  TABLE IF NOT EXISTS `webshop`.`fact_delivery_item` (
  `DELIVERY_ID` INT(11) NOT NULL ,
  `DELIVERY_ITEM_ID` INT(11) NOT NULL ,
  `PRODUCT_ID` INT(11) NOT NULL ,
  `delivery_DATE_ID` INT(11) NOT NULL ,
  `GEO_ID` INT(11) NOT NULL ,
  `MIXED_ID` INT(11) NOT NULL ,
  `delivery_TIME_ID` INT(11) NOT NULL ,
  `Customer_scd_tk` bigINT(11) NOT NULL ,
  `TURNOVER` DECIMAL(10,2) NULL DEFAULT NULL ,
  `QUANTITY` INT(11) NULL DEFAULT NULL ,
  `PROFIT` DECIMAL(10,2) NULL DEFAULT NULL ,
  `RETURNED` VARCHAR(255) NULL DEFAULT NULL ,
  `SHIPPING_COST` DECIMAL(10,2) NULL DEFAULT NULL ,
  `order_TIME_ID` INT(11) NOT NULL ,
  `order_DATE_ID` INT(11) NOT NULL ,
  PRIMARY KEY (`DELIVERY_ID`, `DELIVERY_ITEM_ID`) ,
  INDEX `time_id_idx` (`delivery_TIME_ID` ASC) ,
  INDEX `Product_id_idx` (`PRODUCT_ID` ASC) ,
  INDEX `Date_id_idx` (`delivery_DATE_ID` ASC) ,
  INDEX `Geo_id_idx` (`GEO_ID` ASC) ,
  INDEX `Mixed_id_idx` (`MIXED_ID` ASC) ,
  INDEX `Customer_scd_tkidx` (`Customer_scd_tk` ASC) ,
  INDEX `fk_fact_delivery_item_dim_time1_idx` (`order_TIME_ID` ASC) ,
  INDEX `fk_fact_delivery_item_dim_date1_idx` (`order_DATE_ID` ASC) ,
  CONSTRAINT `Customer_scd_tk`
        FOREIGN KEY (`Customer_scd_tk` )
        REFERENCES `webshop`.`dim_customer` (`Customer_scd_tk` )
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Date_id`
        FOREIGN KEY (`delivery_DATE_ID` )
        REFERENCES `webshop`.`dim_date` (`DATE_ID` )
```

```
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Geo_id`
        FOREIGN KEY (`GEO_ID` )
        REFERENCES `webshop`.`dim_geo` (`GEO_ID` )
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Mixed_id`
        FOREIGN KEY (`MIXED_ID` )
        REFERENCES `webshop`.`dim_mixed` (`MIXED_ID` )
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Product_id`
        FOREIGN KEY (`PRODUCT_ID` )
        REFERENCES `webshop`.`dim_product` (`PRODUCT_ID` )
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `time_id`
        FOREIGN KEY (`delivery_TIME_ID` )
        REFERENCES `webshop`.`dim_time` (`TIME_ID` )
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `order_TIME_ID`
        FOREIGN KEY (`order_TIME_ID` )
        REFERENCES `webshop`.`dim_time` (`TIME_ID` )
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `order_DATE_ID`
        FOREIGN KEY (`order_DATE_ID` )
        REFERENCES `webshop`.`dim_date` (`DATE_ID` )
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

CREATE INDEX idx_dim_customer_lookup
 ON dim_customer
```

```
(
  CUSTOMER_ID
)
;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

A2 SQL Grundlage für Reports

Um die Reports verwenden zu können, müssen zwei Views in MySQL hinzugefügt werden. Zum einen die Umsätze je State(View_1: turnover_year) und die Berechnung zwischen den Jahreswerten (View_2: turnover_year_delta).

Im SQL-Script wird eine Delta-Berechnung zwischen den Jahreszahlen durchgeführt.

MySQL kennt keinen Pivot-Befehl, um Spalten und Zeilen zu vertauschen und entsprechend zu summieren. Durch die Verwendung von Views und einer Anwendung von IF-Bedingungen im SQL-Script (siehe Turnover_Year_Pivot.sql) wurden die Ergebnisse ohne MDX-Abfragen nur mit SQL erstellt. Am Beispiel wird dies für den nächsten Report beschrieben.

```
select dat.YEAR,sum(Turnover) as Turnover_of_state, geo.STATE
from webshop.fact_delivery_item as fact
inner join webshop.dim_geo as geo on fact.GEO_ID = geo.GEO_ID
inner join webshop.dim_date as dat on dat.DATE_ID = fact.order_DATE_ID
Group by geo.STATE, dat.YEAR
Order by geo.State, dat.YEAR
--Ergebnis Umsatz je State je Jahreszahl
```

Abbildung 104 SQL Statement Berechnung Umsatz

Der Umsatz wird je Jahreszahl (2010, 2011, 2012) je State berechnet.

YEAR	Turnover_of_state	STATE
2010	129753.00	California
2011	91399.00	California
2012	103084.00	California
2010	237222.00	Hawaii
2011	243153.00	Hawaii
2012	193635.00	Hawaii
2010	283519.00	Idaho
2011	264450.00	Idaho
2012	308061.00	Idaho
2010	34790.00	Illinois
2011	35624.00	Illinois
2012	25672.00	Illinois

Abbildung 105 Ausgabe Berechnung Umsatz

Als Ergebnis wird eine Liste mit den Umsätzen je Zeile erstellt. Ziel ist es Spalten und Zeilen in SQL zu vertauschen, dies wäre einfach zu lösen, indem ein Pivot-Befehl verwendet werden könnte.

```
Use webshop;
create view turnover_year as (select dat.YEAR as Jahr,sum(Turnover) as Turnover_of_state, geo.STATE
from webshop.fact_delivery_item as fact
inner join webshop.dim_geo as geo on fact.GEO_ID = geo.GEO_ID
inner join webshop.dim_date as dat on dat.DATE_ID = fact.order_DATE_ID
Group by geo.STATE, dat.YEAR
Order by geo.State, dat.YEAR)
--View erstellt
```

Abbildung 106 SQL Statement View Turnover_year

Es werden zusätzlich die Abfragen als Views gespeichert, um später auf die erstellten Views referenzieren zu können.

```
select dat.YEAR, sum(Turnover) from webshop.fact_delivery_item as fact
inner join webshop.dim_geo as geo on fact.GEO_ID = geo.GEO_ID
inner join webshop.dim_date as dat on dat.DATE_ID = fact.order_DATE_ID
Group by dat.YEAR
--Ergebnis Bsp.: Gesamtumsatz: 2010 = 6730317,00
```

Abbildung 107 SQL-Statement Berechnung Gesamtumsatz je Jahr

Um die Prozentwerte berechnen zu können, sollten die Gesamtumsätze berechnet werden, um die einzelnen Umsätze der US-Bundesstaaten zu den Gesamtumsätzen ins Verhältnis setzen zu können.

YEAR	sum(Turnover)
2010	6730317.00
2011	6916811.00
2012	6844933.00

Abbildung 108 Ergebnis Berechnung Gesamtumsatz je Jahr

```
Select STATE,
sum(IF (Jahr = 2010,Turnover_of_state,NULL)) as '2010',
sum(IF (Jahr = 2011,Turnover_of_state,NUll)) as '2011',
sum(IF (Jahr = 2012,Turnover_of_state,Null)) as '2012'
From webshop.turnover_year Group by STATE
--http://en.wikibooks.org/wiki/Programming:MySQL/Pivot_table
```

Abbildung 109 SQL-Statement Pivot-Umwandlung

MySQL unterstützt nicht die Verwendung eines Pivot-Befehls, daher wird eine IF-Bedingung mit einer Sum-Funktion kombiniert.

STATE	2010	2011	2012
California	129753.00	91399.00	103084.00
Hawaii	237222.00	243153.00	193635.00
Idaho	283519.00	264450.00	308061.00
Illinois	34790.00	35624.00	25672.00

Abbildung 110 Ergebnis Pivot-Umwandlung

Für die Delta-Berechnung und die Prozentwerte wurde eine weitere View angelegt.

```
use webshop;
Select *, Round((tyd.2010/s_2010.Gesamtumsatz_2010)*100,2) as inProzent_von_Gesamt_2010,
Round((tyd.2011/s_2011.Gesamtumsatz_2011)*100,2) as inProzent_von_Gesamt_2011,
Round((tyd.2012/s_2012.Gesamtumsatz_2012)*100,2) as inProzent_von_Gesamt_2012
From turnover_year_delta as tyd,
(Select Sum(turnover_year_delta.2010) as Gesamtumsatz_2010 from turnover_year_delta) as s_2010,
(Select Sum(turnover_year_delta.2011) as Gesamtumsatz_2011 from turnover_year_delta) as s_2011,
(Select Sum(turnover_year_delta.2012) as Gesamtumsatz_2012 from turnover_year_delta) as s_2012
--Prozentwerte
```

Abbildung 111 SQL-Statement Delta-Berechnung und Prozentwerte

STATE	2010	Delta_2011_2010	2011	Delta_2012_2011	2012	Gesamtumsatz_2010	Gesamtumsatz_2011	Gesamtumsatz_2012	inProzent_von_Gesamt_2010	inProzent_von_Gesamt_2011	inProzent_von_Gesamt_2012
California	129753.00	-38354.00	91399.00	11685.00	103084.00	6730317.00	6916811.00	6844933.00	1.93	1.32	1.51
Hawaii	237222.00	5931.00	243153.00	-49518.00	193635.00	6730317.00	6916811.00	6844933.00	3.52	3.52	2.83
Idaho	283519.00	-19069.00	264450.00	43611.00	308061.00	6730317.00	6916811.00	6844933.00	4.21	3.82	4.50
Illinois	34790.00	834.00	35624.00	-9952.00	25672.00	6730317.00	6916811.00	6844933.00	0.52	0.52	0.38

Abbildung 112 Ergebnis Delta-Berechnung und Prozentwerte

Die gespeicherten Views sind zum einen die Umsätze je State(turnover_year) und zusätzlich die Berechnung zwischen den Jahreswerten (turnover_year_delta), dargestellt in MySQL.

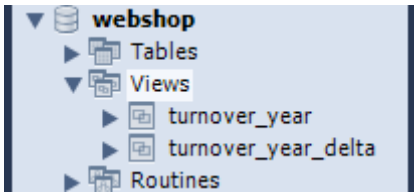


Abbildung 113 angelegte Views der Datenbank Webshop

A3 XML-Struktur Mondrian Schema

```
<Schema name="webshop">
  <Cube name="webshop" visible="true" cache="true" enabled="true">
        <Table name="fact_delivery_item">
        </Table>
        <Dimension type="StandardDimension" visible="true" foreignKey="CUSTOMER_ID" highCardinali-ty="false" name="dim_customer">
        </Dimension>
        <Dimension type="TimeDimension" visible="true" foreignKey="delivery_DATE_ID" highCardinali-ty="false" name="dim_delivery_date">
        <Hierarchy name="default" visible="true" hasAll="true" primaryKey="DATE_ID">
        <Table name="dim_date" alias="">
        </Table>
        <Level name="year" visible="true" column="YEAR" type="Integer" uniqueMembers="false" lev-elType="TimeYears" hideMemberIf="Never">
        </Level>
        <Level name="month" visible="true" column="MONTH" type="Integer" uniqueMembers="false" levelType="TimeMonths" hideMemberIf="Never">
        </Level>
        <Level name="day" visible="true" column="DAY" type="Integer" uniqueMembers="false" lev-elType="TimeDays" hideMemberIf="Never">
        </Level>
        </Hierarchy>
        </Dimension>
        <Dimension type="TimeDimension" visible="true" foreignKey="order_DATE_ID" highCardinali-ty="false" name="dim_order_date">
        <Hierarchy name="default" visible="true" hasAll="true" primaryKey="DATE_ID">
        <Table name="dim_date" alias="">
        </Table>
        <Level name="year" visible="true" column="YEAR" type="Integer" uniqueMembers="false" lev-elType="TimeYears" hideMemberIf="Never">
        </Level>
        <Level name="month" visible="true" column="MONTH" type="Integer" uniqueMembers="false" levelType="TimeMonths" hideMemberIf="Never">
```

```
        </Level>
        <Level name="day" visible="true" column="YEAR" type="Integer" uniqueMem-
bers="false" lev-elType="TimeDays" hideMemberIf="Never">
        </Level>
        </Hierarchy>
        </Dimension>
        <Dimension type="TimeDimension" visible="true" foreignKey="delivery_TIME_ID"
highCardinali-ty="false" name="dim_delivery_time">
        <Hierarchy name="default" visible="true" hasAll="true" prima-
ryKey="TIME_ID">
        <Table name="dim_time" alias="">
        </Table>
        <Level name="hour" visible="true" column="HOUR" type="Integer" unique-
Members="false" lev-elType="TimeHours" hideMemberIf="Never">
        </Level>
        <Level name="minute" visible="true" column="MINUTE" type="Integer"
uniqueMembers="false" levelType="TimeMinutes" hideMemberIf="Never">
        </Level>
        </Hierarchy>
        </Dimension>
        <Dimension type="TimeDimension" visible="true" foreignKey="order_TIME_ID"
highCardinali-ty="false" name="dim_order_time">
        <Hierarchy name="default" visible="true" hasAll="true" prima-
ryKey="TIME_ID">
        <Table name="dim_time" alias="">
        </Table>
        <Level name="hour" visible="true" column="HOUR" type="Integer" unique-
Members="false" lev-elType="TimeHours" hideMemberIf="Never">
        </Level>
        <Level name="minute" visible="true" column="MINUTE" type="Integer"
uniqueMembers="false" levelType="TimeMinutes" hideMemberIf="Never">
        </Level>
        </Hierarchy>
        </Dimension>
        <Dimension type="StandardDimension" visible="true" foreignKey="GEO_ID"
highCardinali-ty="false" name="dim_geo">
        <Hierarchy name="default" visible="true" hasAll="true" primaryKey="GEO_ID">
```

```
        <Table name="dim_geo">
        </Table>
        <Level name="state" visible="true" column="STATE" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="zipcode" visible="true" column="ZIPCODE" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="city" visible="true" column="CITY" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        </Hierarchy>
        </Dimension>
        <Dimension type="StandardDimension" visible="true" foreignKey="PRODUCT_ID" highCardinality="false" name="dim_product">
        </Dimension>
        <Dimension type="StandardDimension" visible="true" foreignKey="MIXED_ID" highCardinality="false" name="dim_mixed">
        </Dimension>
        <Measure name="quantity" column="QUANTITY" aggregator="sum" visible="true">
        </Measure>
        <Measure name="profit" column="PROFIT" aggregator="sum" visible="true">
        </Measure>
        <Measure name="returned" column="RETURNED" aggregator="count" visible="true">
        </Measure>
        <Measure name="shipping_cost" column="SHIPPING_COST" aggregator="sum" visible="true">
        </Measure>
        <Measure name="turnover" column="TURNOVER" aggregator="sum" visible="true">
        </Measure>
  </Cube>
</Schema>
```

A4 Unterschiede zwischen der Enterprise und der Community Edition von Pentaho

Pentaho Community & Enterprise Edition Product Comparison

pentaho
open source business intelligence

	Community Edition	Enterprise Edition
Leading Open Source Business Intelligence	✓	✓
Evaluation		
Free 30 Day Evaluation Support	⊘	✓
Licensing and Pricing		
Low Cost	• Free "Code" • Requires in-house resources to patch and test	• No up-front license costs • Low cost subscription-based pricing
License	• Open Source GPL license • If embedding in a commercial application, obliged to make your application available under the GPL open source license	• Commercial license
Advanced Features		
Automated Installers	⊘	✓
Sophisticated & Interactive Data Analysis, Exploration & Visualization	"Basic" analysis	✓
Self-Service Dashboard Designer	⊘	✓
Hadoop Integration for 'Big Data' Reporting & Analysis	⊘	✓
Support for Mobile BI (e.g. iPad)	⊘	✓
Shared Repository and Team Sharing	⊘	✓
Repository Content Versioning	⊘	✓
Repository Content Locking	⊘	✓
Automated Content Expiration	⊘	✓
Audit Reports	⊘	✓
User and Role-based Permissions	⊘	✓
Security Configuration with LDAP & Active Directory	⊘	✓
Single Sign-On	⊘	✓
Automated ETL Job Scheduling	⊘	✓
Centralized Administration, Diagnostics, and Performance Monitoring	⊘	✓
Clustering	⊘	✓
Test and QA		
Certified and Stable Software	⊘	✓
Deployment		
On Demand and Cloud Hosted Option	⊘	✓
Professional Services		
Classroom / Online Training	⊘	✓
Consulting from Pentaho and Certified Partners	⊘	✓
Ongoing Support		
Community Online Forums	✓	✓
Enterprise Edition Online Forums	⊘	✓
24X7 Enterprise Support with SLAs (Tel/Email)	⊘	✓
Access to Professional Product Documentation	⊘	✓
Remote Assistance Packages (for Installation/Configuration, Design, Troubleshooting/Optimization)	⊘	✓
Knowledge Base of Technical Content	⊘	✓
Release Updates		
Immediate Access to Fully Tested Pentaho Fixes	• Eventually receives fixes (delays of 1-6 months) • Need to re-build code and test manually	✓
Roadmap Visibility and Influence	⊘	✓
Managed Release Cycles	• Unpredictable and constant community code changes	✓
Entitlement to New Features and Upgrades	⊘	✓

Community Edition

Suited for evaluation or pre-production use. Customers choosing this option need internal resources to manage their own release cycles, test, support, and add advanced capabilities to the 'basic' functionality available in this version.

Enterprise Edition

Suited for production deployments. Pentaho BI Suite Enterprise Edition provides advanced, self-contained, and quality-assured software that does not require in-house resources for development and test. Pentaho professional support offers world-class technical support that guarantees resolution times and service level agreements.

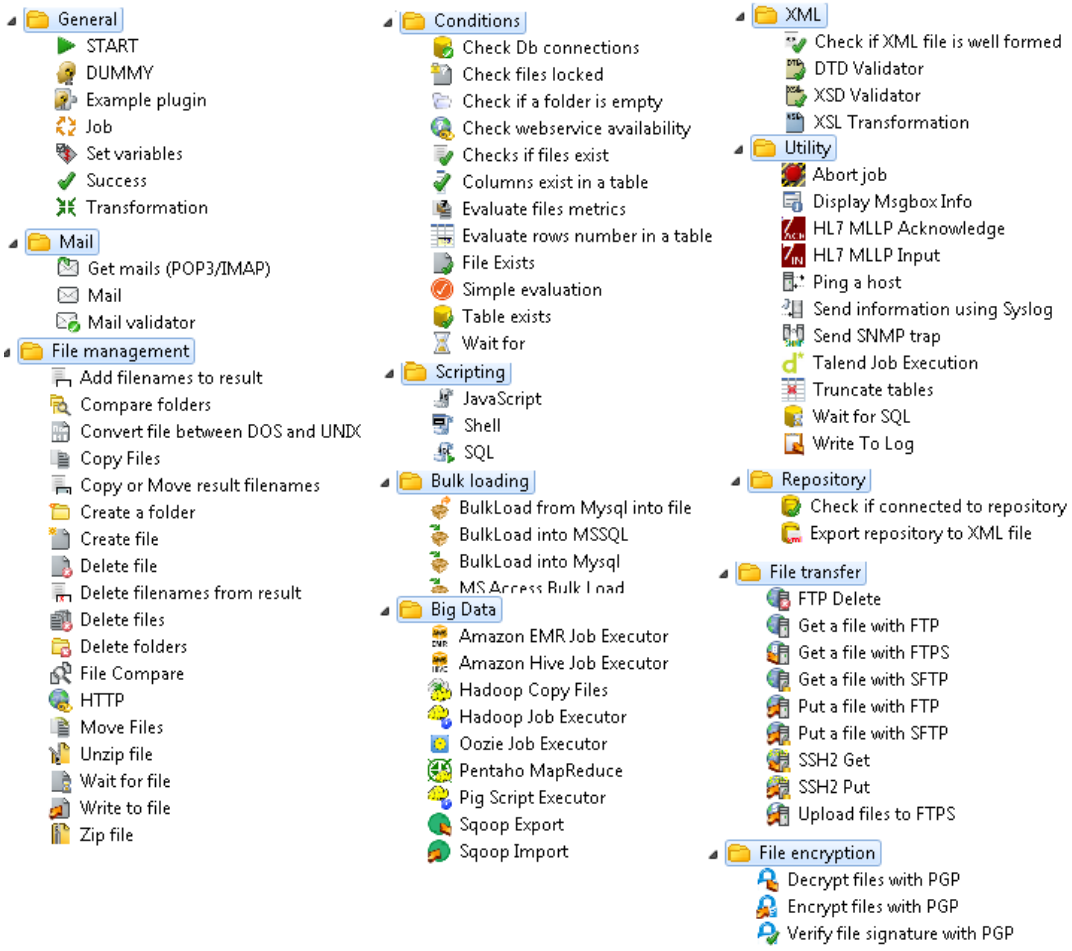
Download Pentaho BI today at:
www.pentaho.com/download

Be Social with Pentaho:

A5 Pentaho Data Integration Übersicht Modellierungsobjekte

Transformation

Job


General
START
DUMMY
Example plugin
Job
Set variables
Success
Transformation
Mail
Get mails (POP3/IMAP)
Mail
Mail validator
File management
Add filenames to result
Compare folders
Convert file between DOS and UNIX
Copy Files
Copy or Move result filenames
Create a folder
Create file
Delete file
Delete filenames from result
Delete files
Delete folders
File Compare
HTTP
Move Files
Unzip file
Wait for file
Write to file
Zip file
Conditions
Check Db connections
Check files locked
Check if a folder is empty
Check webservice availability
Checks if files exist
Columns exist in a table
Evaluate files metrics
Evaluate rows number in a table
File Exists
Simple evaluation
Table exists
Wait for
Scripting
JavaScript
Shell
SQL
Bulk loading
BulkLoad from Mysql into file
BulkLoad into MSSQL
BulkLoad into Mysql
MS Access Bulk Load
Big Data
Amazon EMR Job Executor
Amazon Hive Job Executor
Hadoop Copy Files
Hadoop Job Executor
Oozie Job Executor
Pentaho MapReduce
Pig Script Executor
Sqoop Export
Sqoop Import
XML
Check if XML file is well formed
DTD Validator
XSD Validator
XSL Transformation
Utility
Abort job
Display Msgbox Info
HL7 MLLP Acknowledge
HL7 MLLP Input
Ping a host
Send information using Syslog
Send SNMP trap
Talend Job Execution
Truncate tables
Wait for SQL
Write To Log
Repository
Check if connected to repository
Export repository to XML file
File transfer
FTP Delete
Get a file with FTP
Get a file with FTPS
Get a file with SFTP
Put a file with FTP
Put a file with SFTP
SSH2 Get
SSH2 Put
Upload files to FTPS
File encryption
Decrypt files with PGP
Encrypt files with PGP
Verify file signature with PGP

A6 MDX-Queries

S0_Ausgangssituation

```
SELECT
NON EMPTY {Hierarchize({[dim_delivery_date].[delivery_year].Members})} ON
COLUMNS,
NON EMPTY {Hierarchize({[dim_geo].[state].Members})} ON ROWS
FROM [webshop]
WHERE {[Measures].[profit]}
```

S1_Filter2013

```
SELECT
NON            EMPTY            {Hierarchize({{[dim_delivery_date].[2010],
[dim_delivery_date].[2011], [dim_delivery_date].[2012]}})} ON COLUMNS,
NON EMPTY {Hierarchize({[dim_geo].[state].Members})} ON ROWS
FROM [webshop]
WHERE {[Measures].[profit]}
```

S2_ZeilenFilter40000

```
SELECT
NON            EMPTY            {Hierarchize({{[dim_delivery_date].[2010],
[dim_delivery_date].[2011], [dim_delivery_date].[2012]}})} ON COLUMNS,
NON        EMPTY        Filter({Hierarchize({[dim_geo].[state].Members})},
Measures.Profit < 40000) ON ROWS
FROM [webshop]
WHERE {[Measures].[profit]}
```

S3_WhereFilterType

```
SELECT
NON            EMPTY            {Hierarchize({{[dim_delivery_date].[2010],
[dim_delivery_date].[2011], [dim_delivery_date].[2012]}})} ON COLUMNS,
NON        EMPTY        Filter({Hierarchize({[dim_geo].[state].Members})},
Measures.Profit < 40000) ON ROWS
FROM [webshop]
WHERE                                    CrossJoin({[Measures].[profit]},
{[dim_customer.type_hierarchy].[business]})
```

S4_SortStateProfitDESC

```
SELECT
NON            EMPTY            {Hierarchize({{[dim_delivery_date].[2010],
[dim_delivery_date].[2011], [dim_delivery_date].[2012]}})} ON COLUMNS,
NON        EMPTY        Order({Hierarchize({[dim_geo].[state].Members})},
[Measures].[profit], DESC) ON ROWS
FROM [webshop]
WHERE {[Measures].[profit]}
```

S5_TopPercentStateProfit

```
SELECT
NON EMPTY {Hierarchize({[dim_delivery_date].[2012]})} ON COLUMNS,
NON   EMPTY   TopPercent({Hierarchize({[dim_geo].[state].Members})},   50,
[Measures].[profit]) ON ROWS
FROM [webshop]
WHERE {[Measures].[profit]}
```

S6_PeriodsToDateEditor

```
with
    member dim_delivery_date.Halbjahr1_2010 as aggregate(
PeriodsToDate(
dim_delivery_date.delivery_year, dim_delivery_date.[2010].[6]
))

    member dim_delivery_date.Halbjahr1_2011 as aggregate(
PeriodsToDate(
dim_delivery_date.delivery_year, dim_delivery_date.[2011].[6]
))

    member dim_delivery_date.Halbjahr1_2012 as aggregate(
PeriodsToDate(
dim_delivery_date.delivery_year, dim_delivery_date.[2012].[6]
))

SELECT
NON                                                                 EMPTY
{dim_delivery_date.Halbjahr1_2010,dim_delivery_date.Halbjahr1_2011,dim_de
livery_date.Halbjahr1_2012} ON COLUMNS,
NON EMPTY {Hierarchize({[dim_geo].[state].Members})} ON ROWS
FROM [webshop]
where {[Measures].[profit]}
```